



**Nuno Miguel Aires  
Carriço**

***Audio-guiding* imersivo**



**Nuno Miguel Aires  
Carriço**

***Audio-guiding imersivo***

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor António Guilherme Rocha Campos e coorientação científica do Doutor José Manuel Neto Vieira, Professores Auxiliares do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.



## **o júri**

Presidente

**Professora Doutora Maria Beatriz Alves de Sousa Santos**  
Professor Associado C/Agregação, Universidade de Aveiro

Vogais

**Professor Doutor Marco Paulo Barbosa Conceição**  
Professor Adjunto, Esmae - Escola Superior de Música, Artes e Espectáculo, Instituto Politécnico do Porto (arguente externo)

**Professor Doutor António Guilherme Rocha Campos**  
Professor Auxiliar, Universidade de Aveiro (orientador)

## **agradecimentos**

Gostaria de agradecer ao meu orientador, Professor Guilherme Campos, por toda a dedicação, horas passadas e empenho que teve comigo. Mas acima de tudo, por todos conselhos e motivação que deu ao longo deste percurso.

Ao meu coorientador, Professor José Viera, pelos seus conselhos, disponibilidade e por todo equipamento fornecido necessário à execução deste projeto.

Ao Professor Paulo Dias pela sua disponibilidade e ajuda com familiarização da biblioteca AcousticAVE.

A todos os meus amigos que, de uma forma ou outra, me deram sempre motivação para concluir esta etapa da minha vida.

À minha namorada, Marisa, por toda compreensão e paciência que teve comigo ao longo deste percurso.

Ao meu irmão pela paciência e pelas inúmeras horas em que me ouviu a falar deste trabalho.

Aos meus pais que sempre tudo proporcionaram para que isto fosse possível.

Por fim, a todos que sempre lutaram por mim e nunca desistiram, um enorme Obrigado.

## palavras-chave

*audio-guide*, auralização, *streaming*, processamento, recetor GPS, *head-tracker*, *buffer*, tempo real, HRIR, HRTF, monofónico, binaural, som, fonte sonora virtual, imersivo, realidade aumentada, rota

## resumo

Desenvolveu-se um protótipo de um sistema de audio-guiding que permite associar fontes sonoras virtuais aos pontos focais de cada rota de visita e aplicar, na reprodução estereofónica através de auscultadores, um efeito de espacialização controlado em tempo real pela pose (posição e orientação da cabeça) do utilizador relativamente a esses pontos. É assim criado um efeito de realidade aumentada, com os conteúdos áudio (virtuais) a parecerem provir de pontos (reais) especificados.

Foi desenvolvido um programa de gestão de rotas para especificar previamente os pontos focais (através de coordenadas GPS), o conteúdo áudio (ficheiro monofónico) pretendido para cada e o ponto do trajeto em que deve ser iniciada a reprodução.

Para detetar posição, usou-se um recetor GPS ligado à plataforma de computação transportada pelo utilizador; para monitorizar continuamente os movimentos da sua cabeça (*head-tracking*), recorreu-se a um sensor inercial (*InertiaCube*) acoplado aos auscultadores.

A aplicação principal, desenvolvida em linguagem C++, recorre a buffers circulares implementados em software para realizar, com a mínima latência possível, o streaming áudio a partir dos ficheiros de entrada. O processamento baseia-se numa biblioteca ('motor') de auralização em tempo real que utiliza bases de dados de HRTF de acesso público. Para escolher o par de HRTF a utilizar em cada ciclo de processamento, o azimute e a elevação da fonte virtual são continuamente recalculados em função dos dados recolhidos sobre a pose do utilizador.

Para avaliar o funcionamento da aplicação, identificar possíveis problemas e caracterizar as gamas mais adequadas para determinados parâmetros (e.g. tamanho de buffers, número de amostras por bloco de áudio, janela de filtragem dos dados GPS), efetuaram-se testes subjetivos preliminares, com dez sujeitos a percorrer uma rota criada para o efeito. Embora confirmando a capacidade de obter o efeito sonoro pretendido, os testes evidenciaram a necessidade de melhorar a precisão dos dados de posição, principal fator a afetar negativamente a qualidade da experiência.

**keywords**

audio-guide, auralization, streaming, processing, GPS receiver, head-tracker, buffer, real-time, HRIR, HRTF, monophonic, binaural, sound, virtual sound source, immersive, augmented reality, route

**abstract**

An audio-guiding system prototype was developed which makes it possible to associate virtual sound sources to the focal points of each tourist route and apply, in the stereophonic reproduction over headphones, a spatialisation effect controlled in real time by user pose (position and head orientation) relative to those points. An augmented reality effect is thus achieved, with the (virtual) audio content seemingly originating from specified (real) points.

A route management program was developed to allow specification of the focal points (through GPS coordinates), audio content (monophonic file) intended for each and route point where its playback should be triggered.

Position was detected by a GPS receptor plugged into the computing platform carried by the user; head-tracking was based on an inertial sensor (InertiaCube) attached to the headphone strap.

The main application, developed in C++, implemented software buffers to stream audio through from the input files with the lowest possible latency. The processing resorts to a real-time auralisation engine using a public access HRTF database. In order to choose the appropriate HRTF pair for each processing cycle, the azimuth and elevation of the virtual source are continuously recalculated according to the acquired user pose data.

In order to assess performance, identify possible problems and characterise the most appropriate application parameter ranges (e.g. buffer size, number of samples per audio block, filtering window for GPS data), preliminary subjective tests were carried out, with ten subjects following a route created for this purpose. Whilst confirming the ability to provide the desired audio spatialisation effects, the tests also evidenced the need to improve the precision of position data, as this was the main factor negatively affecting the experience.





# Conteúdo

<b>1. Introdução .....</b>	<b>1</b>
1.1 Motivação .....	1
1.2 Sistemas de <i>audio-guiding</i> .....	3
1.3 Visão global do sistema .....	9
1.4 Estrutura da dissertação .....	11
<b>2. Sistema de Auralização .....</b>	<b>13</b>
2.1 Introdução.....	13
2.1.1 Auralização.....	13
2.1.2 Som.....	13
2.1.3 Anatomia do ouvido humano .....	15
2.1.4 Audição binaural .....	16
2.1.5 Perceção acústica em salas .....	19
2.1.6 AcousticAVE.....	21
2.2 Aplicações .....	24
2.2.1 Auralização com <i>streaming</i> a partir de microfone.....	25
2.2.2 Auralização com <i>streaming</i> a partir de ficheiro áudio.....	28
<b>3. Sistemas de deteção de posição e orientação .....</b>	<b>32</b>
3.1 Deteção de posição .....	32
3.1.1 Introdução.....	32
3.1.2 Recetor GPS .....	35
3.1.3 Aplicação para aquisição de dados do recetor GPS .....	37
3.2 Deteção de orientação.....	40
3.2.1 Introdução.....	40
3.2.2 Sensor inercial .....	41
3.2.3 Aplicação para aquisição de dados do sensor inercial .....	41
<b>4. Integração dos sistemas .....</b>	<b>44</b>
4.1 Gestor de rota .....	44
4.2 Controlo de reprodução .....	47
4.2.1 Implementação e funcionamento .....	48
4.2.2 Interface da aplicação .....	56
<b>5. Testes e validação.....</b>	<b>59</b>
5.1 Conceção e preparação dos testes .....	59
5.2 Resultados .....	62
5.3 Análise crítica dos resultados .....	63
<b>6. Conclusões e trabalho futuro .....</b>	<b>66</b>

<b>Anexo A - <i>Software</i> .....</b>	<b>69</b>
<b>Anexo B - Bibliotecas e <i>device drivers</i> .....</b>	<b>71</b>
<b>Anexo C - Conteúdo do CD .....</b>	<b>73</b>
<b>Anexo D - Compilação e parâmetros configuráveis .....</b>	<b>75</b>
<b>Referências .....</b>	<b>79</b>



## Lista de Figuras

Figura 1-1: Sistemas de <i>audio-guiding</i> .....	1
Figura 1-2: Fontes virtuais e zonas de <i>trigger</i> em rota com <i>audio-guiding</i> imersivo .....	2
Figura 1-3: Arquitetura típica de um sistema de <i>audio-guiding</i> .....	5
Figura 1-4: Unidade central de distribuição de áudio (fonte: <a href="http://www.toguide.pt/">http://www.toguide.pt/</a> ) .....	5
Figura 1-5: Consola de utilizador em autocarro <i>hop on hop off</i> (fonte: <a href="https://hopon-hopoff.vn/">https://hopon-hopoff.vn/</a> ) .....	6
Figura 1-6: Dispositivos de <i>audio-guiding</i> portáteis: emissor (à esquerda) e recetor (à direita) .....	6
Figura 1-7: Sistema de <i>audio-guiding</i> centrado nos terminais .....	7
Figura 1-8: Arquitetura do sistema pretendido em autocarros <i>hop on hop off</i> .....	8
Figura 1-9: Arquitetura do sistema pretendido adaptada a <i>smartphone</i> .....	8
Figura 1-10: Diagrama global do sistema .....	9
Figura 1-11: Composição do bloco de reprodução .....	10
Figura 1-12: Auralização e <i>streaming</i> de áudio .....	10
Figura 2-1: Onda longitudinal [10] .....	14
Figura 2-2: Radiação com origem em um ponto (ilustração 2D) [10] .....	14
Figura 2-3: Frente de onda esférica em relação ao seu raio .....	15
Figura 2-4: Anatomia do ouvido humano [10] .....	15
Figura 2-5: Manequim de aquisição de HRTF (fonte: <a href="https://www.ece.ucdavis.edu/cipic/spatial-sound/research/">https://www.ece.ucdavis.edu/cipic/spatial-sound/research/</a> ) .....	17
Figura 2-6: a) Sistema de coordenadas; b) Projeções dos ângulos de azimute e elevação .....	17
Figura 2-7: Esquema de localização dos pontos de emissão na medição de HRTF a) vista frontal b) vista lateral (fonte: <a href="https://www.ece.ucdavis.edu/cipic/files/2015/04/cipic_CIPIC_HRTF_Database.pdf">https://www.ece.ucdavis.edu/cipic/files/2015/04/cipic_CIPIC_HRTF_Database.pdf</a> ) .....	18
Figura 2-8: Processo de auralização binaural [10] .....	18
Figura 2-9: Medição da RIR em uma sala .....	20
Figura 2-10: Resposta impulsional de uma sala genérica .....	20
Figura 2-11: AcousticAVE: processo de auralização .....	22
Figura 2-12: Fonte sonora e sua imagem virtual definindo a direção de uma reflexão de primeira ordem .....	22
Figura 2-13: Primeira aplicação de ensaio do sistema de auralização .....	25
Figura 2-14: <i>Buffer</i> circular e posicionamento dos ponteiros de escrita e leitura .....	25
Figura 2-15: Fluxograma da primeira aplicação de ensaio .....	26
Figura 2-16: Fluxogramas de funcionamento das funções de <i>callback</i> : a) de gravação; b) de reprodução .....	27

Figura 2-17: Segunda aplicação de ensaio do sistema de auralização .....	28
Figura 2-18: Fluxograma da segunda aplicação de ensaio .....	29
Figura 2-19: Alocação e inicialização de memória para o áudio de entrada .....	30
Figura 3-1: Triangulação com satélites [24] .....	33
Figura 3-2: Sistema de coordenadas geográficas: ângulos de longitude ( $\lambda$ ) e latitude ( $\phi$ ) [24] .....	33
Figura 3-3: Convenções na medida dos ângulos de longitude e latitude (fonte: <a href="https://www.e-education.psu.edu/">https://www.e-education.psu.edu/</a> ) .....	34
Figura 3-4: Recetor GPS Xucai GD75 (fonte: <a href="https://pt.aliexpress.com/">https://pt.aliexpress.com/</a> ) .....	35
Figura 3-5: <i>Screenshot</i> da receção de dados do receptor GPS com PuTTY .....	37
Figura 3-6: a) Interrupção da porta série; b) <i>Thread</i> de receção de dados de GPS ( <i>Thread B</i> ) .....	39
Figura 3-7: Interface da aplicação de aquisição de dados do recetor GPS .....	39
Figura 3-8: Rotações em relação aos eixos coordenados .....	40
Figura 3-9: <i>InertiaCube3</i> da <i>Intersense</i> (fonte: <a href="https://est-kl.com/it/manufacture/intersense/wireless-inertiacube-3.html">https://est-kl.com/it/manufacture/intersense/wireless-inertiacube-3.html</a> ) .....	41
Figura 3-10: Funcionamento da aplicação de aquisição de dados do sensor inercial .....	42
Figura 3-11: Aplicação de aquisição de dados do sensor <i>InertiaCube3</i> .....	42
Figura 4-1: Ficheiro de rota .....	44
Figura 4-2: Interface gráfica da aplicação de gestão de rota .....	46
Figura 4-3: Elementos integrados na aplicação de reprodução e suas trocas de informações .....	47
Figura 4-4: Diagrama temporal de execução das <i>threads</i> e principais trocas de informação entre elas .....	48
Figura 4-5: Seleção do modo de <i>trigger</i> .....	50
Figura 4-6: Modo 1 de <i>trigger</i> .....	51
Figura 4-7: Modo 2 de <i>trigger</i> .....	52
Figura 4-8: Fluxo de dados na <i>thread</i> de auralização e <i>streaming</i> de áudio .....	53
Figura 4-9: Posições relativas do ouvinte e da fonte sonora .....	54
Figura 4-10: a) Cálculo de $L(x)$ b) Cálculo de $L(y)$ .....	55
Figura 4-11: Interface principal da aplicação da <i>Immersive Audio guiding</i> .....	56
Figura 4-12: Janela do painel de configuração .....	56
Figura 4-13: a) Janela da posição do ouvinte b) Janela da orientação da cabeça do ouvinte .....	57
Figura 4-14: Janela de estado das fontes .....	57
Figura 5-1: Traçado da rota de teste .....	59
Figura 5-2: Equipamento de teste .....	61



## Lista de Tabelas

Tabela 3-1: Recetores GPS e suas características principais .....	35
Tabela 3-2: Características do recetor de GPS .....	35
Tabela 3-3: Formatação da informação na mensagem \$GPGGA [27] .....	36
Tabela 3-4: Parâmetros da comunicação série do recetor GPS .....	37
Tabela 3-5: Caraterísticas do sensor <i>InertiaCube3</i> .....	41
Tabela 5-1: Configuração da rota de teste .....	60
Tabela 5-2: Ficheiros áudio da rota de teste .....	60
Tabela 5-3: Chave de quantificação dos resultados .....	62
Tabela 5-4: <i>Trigger</i> a uma distância aparentemente correta .....	62
Tabela 5-5: Sensação do som provir da localização correta .....	62
Tabela 5-6: Repetição de reprodução sem reentrada .....	62





# Lista de Acrónimos

<b>AcousticAVE</b>	<i>Auralisation Models and Applications for Virtual Reality Environment</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>ASCII</b>	<i>American Standard Code for Information Interchange</i>
<b>CAD</b>	<i>Computer Aided Design</i>
<b>CR</b>	<i>Carriage Return</i>
<b>FIFO</b>	<i>First In First Out</i>
<b>GPGBA</b>	<i>Global Positioning System Fixed Data</i>
<b>GPS</b>	<i>Global Positioning System</i>
<b>GUI</b>	<i>Graphical User Interface</i>
<b>HRIR</b>	<i>Head-related Impulse Response</i>
<b>HRTF</b>	<i>Head-related Transfer Function</i>
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>IID</b>	<i>Interaural Intensity Difference</i>
<b>IMU</b>	<i>Inertial Measurement Unit</i>
<b>ITD</b>	<i>Interaural Time Difference</i>
<b>LBS</b>	<i>Location-Based Service</i>
<b>LF</b>	<i>Line Feed</i>
<b>LTI</b>	<i>Linear Time-Invariant</i>
<b>MIS</b>	<i>Mirror-Image Source</i>
<b>NMEA</b>	<i>National Marine Electronics Association</i>
<b>QR</b>	<i>Quick Response</i>
<b>RA</b>	<i>Realidade Aumentada</i>
<b>RFID</b>	<i>Radio-Frequency IDentification</i>
<b>RIR</b>	<i>Room Impulse Response</i>
<b>RV</b>	<i>Realidade Virtual</i>

**USB**

*Universal Serial Bus*

# 1. Introdução

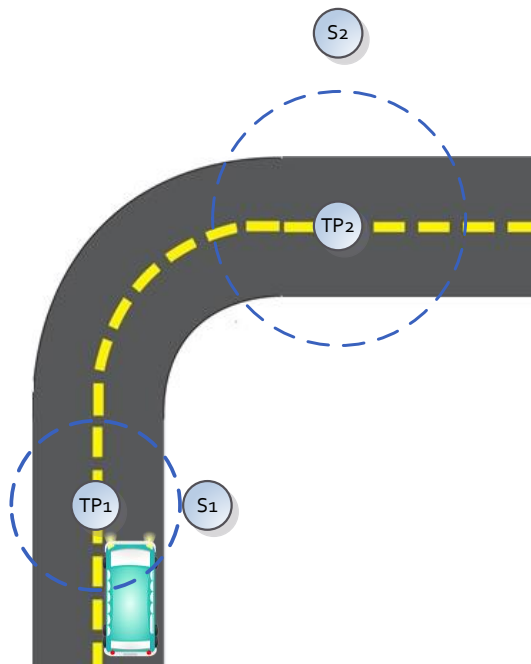
## 1.1 Motivação

Os sistemas de *audio-guiding* (Figura 1-1) têm-se popularizado de forma acentuada nos últimos anos. Trata-se de sistemas eletrónicos que disponibilizam conteúdos áudio para guiar visitas, normalmente de carácter turístico. Muitas vezes, os *audio-guides* são usados em visitas turísticas simplesmente para facilitar a comunicação por parte do guia, que usa um microfone para transmitir ao seu grupo informação (histórica, técnica, artística...) *in vivo*. Outros funcionam de forma autónoma, reproduzindo, de acordo com a posição na rota turística, conteúdos pré-gravados com o mesmo tipo de informação. Sistemas deste último tipo encontram-se cada vez mais frequentemente disponíveis em museus, monumentos, sítios históricos ou postos de turismo, bem como a equipar veículos turísticos (autocarros, embarcações...). Os autocarros *hop-on hop-off* constituem talvez a aplicação mais típica destes sistemas. Podem ter particular utilidade em ambientes ruidosos (e.g. urbanos ou industriais) e para visitantes com capacidade auditiva diminuída.



**Figura 1-1:** Sistemas de *audio-guiding*

Este projeto tem em vista enriquecer a experiência imersiva que os *audio guides* podem proporcionar, tornando-os capazes de criar ambientes de realidade aumentada (RA) áudio. Pretende-se desenvolver um protótipo que permita associar fontes sonoras virtuais (conteúdos áudio gravados para o efeito) a pontos de interesse turístico em ambiente exterior (por exemplo, edifícios ou estátuas) e, através de auscultadores, transmitir a sensação de que os conteúdos áudio são emitidos a partir desses pontos ('S' - *sources*). Para isso, o protótipo deve permitir também definir as zonas do trajeto do ouvinte (em torno de pontos 'TP' - *trigger points*) em que deve ser reproduzido o ficheiro áudio associado a cada fonte virtual. A Figura 1-2 ilustra o funcionamento pretendido.



**Figura 1-2:** Fontes virtuais e zonas de *trigger* em rota com *audio-guiding* imersivo

Este projeto irá recorrer a *software* de auralização em tempo real que processa o material monofónico de cada fonte por meio de filtros *Head-Related Transfer Function* (HRTF), escolhidos em função da posição e da orientação da cabeça do ouvinte em relação à fonte virtual, produzindo assim áudio binaural com o efeito de espacialização 3D pretendido. Com este fim, o protótipo integra vários sistemas, nomeadamente um recetor de *Global Positioning System* (GPS), que fornece a localização do ouvinte em cada instante, e um dispositivo de *head-tracking* acoplado aos auscultadores, para detetar a orientação espacial da cabeça.

## 1.2 Sistemas de *audio-guiding*

Numa visita guiada convencional, um guia apresenta os conteúdos aos ouvintes em ambiente acústico natural, sem intermediação de qualquer sistema eletrónico.

Os sistemas de *audio-guiding* podem ser úteis em diferentes cenários e com múltiplas finalidades, o que levou ao desenvolvimento de uma grande variedade de marcas e modelos, permitindo as mais diversas configurações. A funcionalidade extra que se pretende introduzir neste trabalho (espacialização 3D) foi já discutida na secção anterior.

Antes da conceção de um novo sistema ou dispositivo ou da introdução de uma nova *feature* (como é o caso) é indispensável fazer um estudo sobre os sistemas que já existem e como atuam, para definir em que sentido e de que forma podem ser aperfeiçoados. Procedeu-se por isso a uma revisão da literatura, quer na vertente académica (estudos sobre a tipologia e características de sistemas propostos ou já existentes) quer na vertente comercial (sistemas disponíveis no mercado, modelos mais popularizados e suas funcionalidades).

A Tamo apresenta modelos como o GPS Multilingual Commentary System [1], usado em autocarros *hop-on hop-off* que apresenta uma arquitetura com um elemento central (servidor áudio) que permite receber sinal através de um microfone (*wireless* ou cablado) ou reproduzir conteúdos pré-gravados. Esta última opção pode ser ativada de forma manual (pela guia), ou automática, com recurso a um sistema de *trigger* (acoplado ao servidor áudio) controlado por GPS (o que implica uma rota pré-definida). Suporta conteúdos em várias línguas que são difundidos por cabo até aos terminais, onde cada ouvinte liga os seus auscultadores e seleciona os conteúdos pretendidos.

Há vários outros sistemas usados em autocarros *hop-on hop-off* baseados na mesma arquitetura, com pequenas variantes. Uma delas refere-se à forma como o sinal é difundido entre o servidor áudio e os terminais de cada ouvinte. Por exemplo, os sistemas TriggerPoint Wireless [2] da toGuide recorrem a *wifi*, evitando assim cabos. Também podemos encontrar arquiteturas que usam sistemas de *trigger* diferentes. É o caso do Bus Audio Guide System [3] da Soolai que recorre à tecnologia de *Radio-Frequency IDentification* (RFID). Isto representa uma desvantagem em relação ao GPS na medida em que implica uma preparação prévia dos percursos. Mas, por outro, lado é utilizável indiferentemente em ambientes interiores e exteriores.

Para situações em que o ouvinte necessite de se deslocar a pé, como é o caso de museus, exposições ou percursos pedestres ao ar livre, utilizam-se dispositivos portáteis específicos ou aplicações para *smartphone/tablet*.

O modelo WTG-500 Tour Guide System [4] da Takstar é um exemplo da primeira situação. A sua arquitetura caracteriza-se por ter um dispositivo transmissor (conceitualmente o servidor áudio) que acompanha o guia e que emite, via rádio, os conteúdos captados por um microfone usado pelo guia. Cada ouvinte tem consigo um dispositivo recetor (onde são acoplados os auscultadores) que permite regular o volume e selecionar o canal (banda de frequência) em que são transmitidos esses conteúdos. Isto possibilita o funcionamento de vários grupos em simultâneo (um transmissor para múltiplos recetores), permitindo aos utilizadores ouvirem guias diferentes para, por exemplo, ouvirem línguas distintas ou outro tipo de conteúdos.

A Mix Tech Polska apresenta os modelos ATSG02 [5] ou ATGS03 [6], que recorrem a conteúdos pré-gravados nos dispositivos que acompanham os ouvintes (permitindo desta maneira várias línguas), não

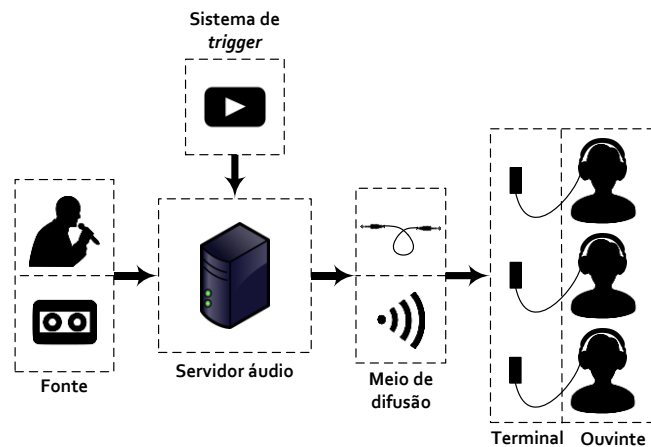
necessitando assim de um guia e consequentemente de dispositivos transmissores de conteúdos. Naturalmente, estes dispositivos necessitam que haja um *trigger* para reprodução. O ATSG02 apenas permite um *trigger* manual através da digitação de um código (em teclado numérico presente no dispositivo) em função do objeto de que se pretende obter informação. O modelo ATSG03 apresenta, além do modo manual (igual ao do ATSG02), um modo automático com recurso a transmissores de sinal (nos objetos da visita) que servem para ativar a reprodução dos conteúdos por deteção de proximidade.

Como exemplo de aplicações para *smartphone/tablet* destinadas a visitas guiadas, podemos referir as da Acoustiguide [7] que cobrem ambientes *outdoor* (e.g. ‘A Walk Through Gilded NY’) e *indoor* (e.g. ‘9/11 Museum Audio Guide’). Contemplam os sistemas operativos Android e iOS, permitem conteúdos em várias línguas e possibilitam obter os conteúdos áudio quer descarregando-os antecipadamente para o dispositivo quer por *streaming*, através de *wifi* ou serviços de dados móveis. Existe a possibilidade de *trigger* manual (feito pelo ouvinte) através da leitura de um código *Quick Response* (QR) por seleção na aplicação do conteúdo desejado, ou automático, com base em tecnologias de *wifi positioning*, *Bluetooth* ou GPS (este exclusivo para ambientes *outdoor*).

A pesquisa realizada apenas permitiu identificar um sistema comercial – o Usomo [8] – que proporciona um efeito de espacialização sonora. Este sistema, destinado exclusivamente a ambientes *indoor*, compreende uma aplicação para *smartphone* e *headphones* equipados com um módulo de *tracking* desenvolvido pela marca. Este assegura o *trigger* dos conteúdos e a monitorização da distância e orientação dos ouvintes em relação às fontes sonoras virtuais, com base em transmissores *wireless* distribuídos pelo espaço destinado à visita. O *smartphone* armazena os conteúdos áudio. O sistema proporciona assim uma experiência imersiva 360° através de som binaural gerado em tempo real. No entanto, não cobre as aplicações visadas nesta dissertação, já que se pretende um sistema destinado a ambientes *outdoor* com *trigger* baseado em GPS, evitando assim a necessidade de instalar transmissores nos elementos da rota de visita.

Pode especular-se sobre a razão do aparente desinteresse pela comercialização de um sistema de *audio guide* com espacialização sonora para ambientes *outdoor*. O custo de implementação constituirá, provavelmente, o principal fator. Além do custo intrínseco do sistema (*head-tracking*, deteção de posição e processamento em tempo real), pode haver necessidade de investimento adicional. Numa aplicação típica de *audio-guides* em autocarro *hop-on hop-off* são normalmente fornecidos auscultadores descartáveis. Ora, para que a mais-valia do efeito de espacialização sonora não seja anulada por ruídos externos (e.g. motor do veículo, ambiente urbano, conversa entre outros visitantes...) podem ser indispensáveis auscultadores de qualidade mais elevada e até cancelamento ativo de ruído.

Os sistemas de *audio-guiding* existentes, de uma maneira geral, podem ser divididos em 6 blocos, como ilustra a Figura 1-3, ou menos, por combinação de blocos.



**Figura 1-3:** Arquitetura típica de um sistema de *audio-guiding*

Em referência à Figura 1-3, podemos identificar:

- **Fonte** - Guia *in vivo* ou conteúdos áudio pré-gravados.
- **Servidor áudio** - Dispositivo de *hardware* que difunde o áudio pelos ouvintes.
- **Sistema de trigger** - Elementos que controlam o início da reprodução, nos casos em que são usados conteúdos áudio pré-gravados.
- **Meio de difusão** - Por cabo ou via rádio (por exemplo *wi-fi*).
- **Terminal** - Consola ou dispositivo portátil (e.g. *smartphone*) usado para ligar os auscultadores, escolher o canal de difusão, configurar o volume ou alterar outras configurações.
- **Ouvinte** - Utilizador do sistema, equipado com *earphones* ou *headphones*.

Os sistemas usados em veículos *hop-on hop-off* (e.g. autocarros, barcos...) obedecem perfeitamente divisão em blocos apresentada na Figura 1-3. Têm um servidor áudio central fixo (vide exemplo na Figura 1-4) que distribui os conteúdos áudio por vários terminais (dentro do veículo, podendo ser fixos aos bancos dos veículos ou portáteis), onde os utilizadores ligam os seus auscultadores.



**Figura 1-4:** Unidade central de distribuição de áudio (fonte: <http://www.toguide.pt/>)

O sinal áudio que essa unidade distribui pode provir de um microfone (guia *in vivo*) ou ter sido previamente gravado e associado a um traçado de rota. Só nesta segunda situação é necessário controlar o início da reprodução, o que pode ser feito manualmente (pelo próprio guia, por exemplo) ou por um sistema automático de *trigger* (baseado em GPS, por exemplo).

Normalmente existe um terminal por cada ouvinte (vide Figura 1-5), permitindo configuração individualizada de vários parâmetros (língua, volume,...).



**Figura 1-5:** Consola de utilizador em autocarro *hop on hop off* (fonte: <https://hopon-hopoff.vn/>)

Também existem sistemas indicados para visitas pedestres, em geral adaptados tanto a ambientes *outdoor* (por exemplo, visita de parques e monumentos) como *indoor* (por exemplo, museus e galerias de arte). Neste caso, os recetores tornam-se dispositivos portáteis sem fios transportados pelos ouvintes. Tipicamente, o guia utiliza um dispositivo emissor (conceptualmente o servidor de áudio, que passa também a ser portátil). No exemplo da Figura 1-6, o emissor e o recetor têm o mesmo aspecto físico. Podem utilizar-se (normalmente para efeito de tradução simultânea) dispositivos a funcionar simultaneamente como recetores e emissores (em canais diferentes), podendo os ouvintes seleccionar o canal pretendido.

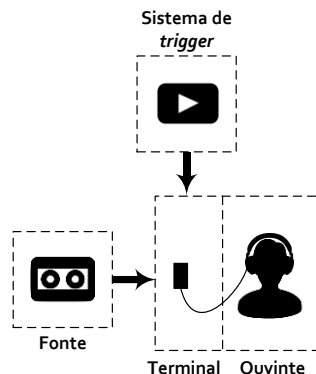


**Figura 1-6:** Dispositivos de *audio-guiding* portáteis: emissor (à esquerda) e recetor (à direita)

(fonte: <http://www.takstar.com/>)



As versões vocacionadas exclusivamente para conteúdos pré-gravados apresentam uma arquitetura bastante simplificada, como mostra a Figura 1-7, uma vez que os recetores podem centralizar todas as funções do bloco servidor de áudio: armazenamento de conteúdos, ligação dos auscultadores e sistema de trigger.



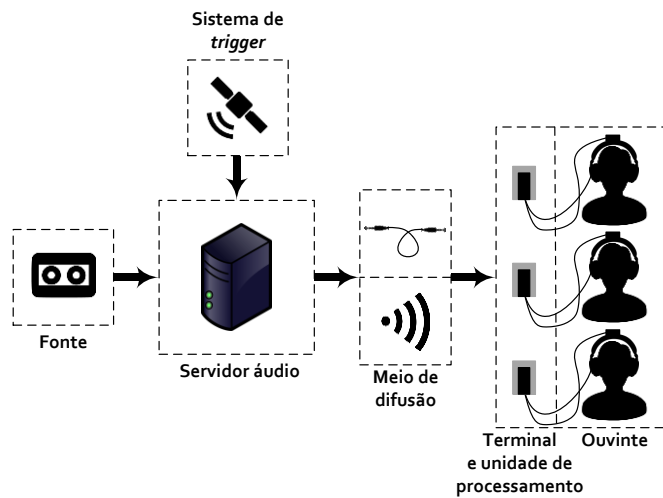
**Figura 1-7:** Sistema de *audio-guiding* centrado nos terminais

Os conteúdos são carregados diretamente nos próprios recetores ou em memória externa acoplada (e.g. *flash drives* com conexão *Universal Serial Bus* - USB). Os sistemas de *trigger* passam a estar integrados nos terminais, usando frequentemente tecnologias de infravermelhos ou *Radio-Frequency IDentification* - RFID.

Existem variantes com reprodução em alta-voz, permitindo que o mesmo aparelho seja utilizado por vários visitantes ao mesmo tempo.

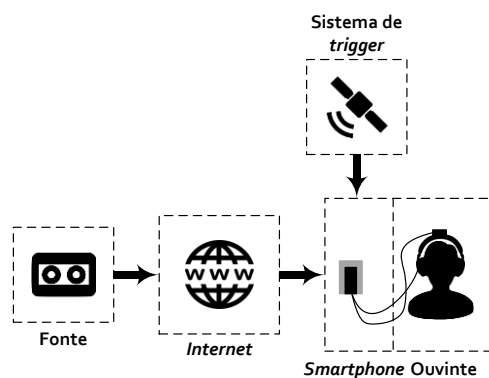
Os sistemas baseados em aplicações (*apps*) em *smartphone* ou *tablet* também recorrem ao mesmo tipo de configuração, com conteúdos pré-gravados sem necessidade de difusão de conteúdos áudio, podendo-se considerar que os dispositivos são servidores e terminais em simultâneo. São bastante versáteis, uma vez que também podem ser usados em ambientes *indoor* ou *outdoor* e as aplicações podem ser descarregadas da *internet* sempre que houver necessidade. Os conteúdos podem estar guardados em memória ou ser descarregados da *internet* em tempo real. O *trigger* pode ser manual (selecção directa do ouvinte na aplicação), semiautomático (código QR), ou automático, por algoritmos de localização com base na rede móvel, GPS, *wi-fi*, *Location-Based Services* (LBS), que combinam vários dos anteriores para uma maior precisão, ou outras tecnologias.

Tendo em conta esta análise e dado o foco deste trabalho ser a utilização em autocarros *hop-on hop-off*, o sistema pretendido deve basear-se numa arquitetura que use conteúdos pré-gravados e detecção da localização do ouvinte através de GPS, e integrar um novo elemento: um dispositivo de *head-tracking* para monitorizar os movimentos da cabeça de cada ouvinte. Cada terminal deve agora ter capacidade de processar o sinal de áudio em tempo real de modo a criar o efeito de espacialização sonora 3D, que será, em geral, diferente para cada ouvinte, pois os movimentos das suas cabeças são independentes um dos outros. A Figura 1-8 enquadra esta especificação na estrutura geral da Figura 1-3. Pode manter-se o sistema de *trigger* centralizado, pois a reprodução dos conteúdos deve ser simultânea para todos os ouvintes.



**Figura 1-8:** Arquitetura do sistema pretendido em autocarros *hop on hop off*

Como hoje em dia os *smartphones* apresentam uma grande panóplia de sensores e grande capacidade de processamento e memória, torna-se claro que o protótipo a desenvolver poderá ser facilmente adaptado para usar um *smartphone*, como sugere a Figura 1-9. O *trigger* pode ser baseado no recetor GPS do próprio dispositivo. Apesar de os *smartphones* integrarem giroscópios, é necessário um dispositivo de *head-tracking* dedicado, pois o telemóvel não estará acoplado à cabeça do ouvinte.

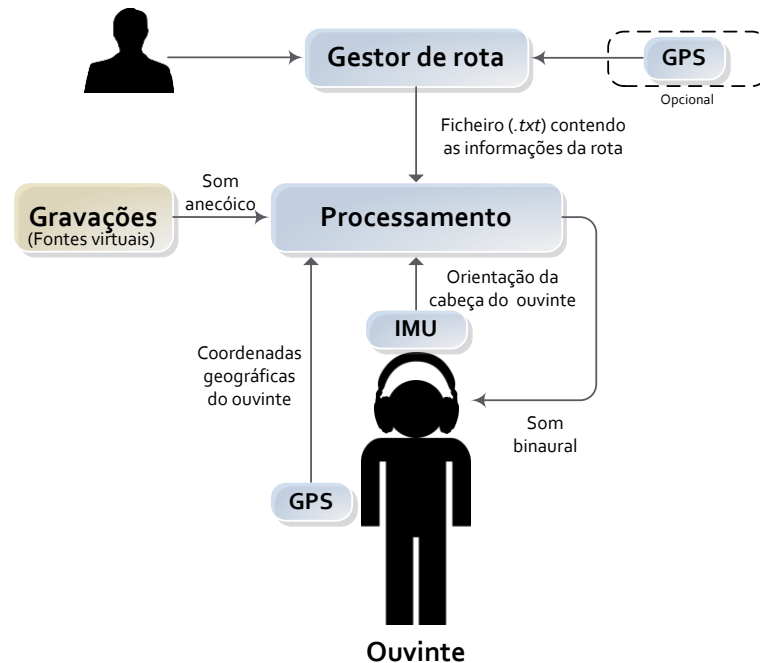


**Figura 1-9:** Arquitetura do sistema pretendido adaptada a *smartphone*

Parece claro que este trabalho pode trazer algo de novo relativamente aos sistemas existentes, com a introdução de uma *feature* até hoje inexplorada – a capacidade de criar um efeito sonoro 3D, tornando a experiência do ouvinte mais imersiva.

### 1.3 Visão global do sistema

Este projeto exige a interligação e integração de vários subsistemas, representados no esquema da Figura 1-10 pelos blocos ‘IMU’, ‘GPS’, ‘Gestor de rota’ e ‘Reprodução’, brevemente descritos a seguir.



**Figura 1-10:** Diagrama global do sistema

- **IMU**

Trata-se de um sensor inercial (*Inertial Measurement Unit*) instalado de forma a monitorizar os movimentos da cabeça do ouvinte (*head-tracking*). Os dados obtidos são fornecidos ao bloco ‘Reprodução’.

- **GPS**

É um dispositivo que assegura a receção de sinais de GPS. Associado ao ouvinte, permite fazer um rastreamento contínuo da sua posição (coordenadas geográficas) e fornecê-la ao bloco ‘Reprodução’.

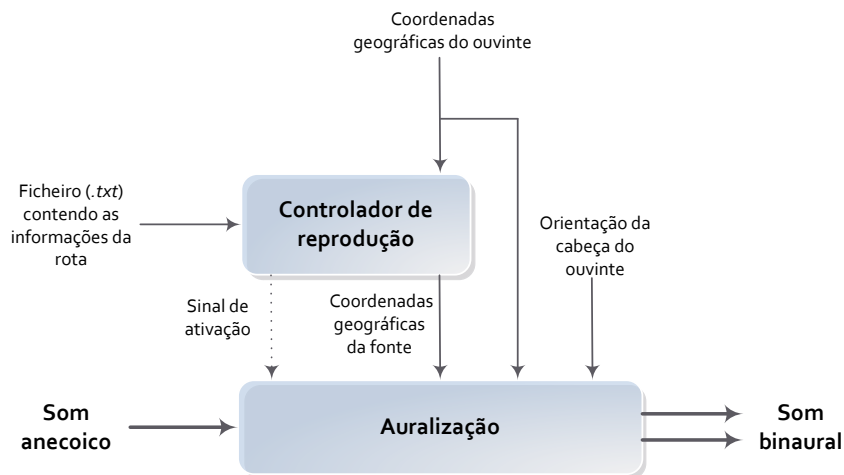
- **Gestor de rota**

Este bloco permite criar uma coleção de trilhas áudio monofónicas (gravações anecóicas), cada uma associada a uma fonte virtual. Permite também definir as coordenadas geográficas quer da própria fonte, quer da região em que se pretende ativar a reprodução (região de *trigger*). É constituído por uma aplicação, em ambiente Windows, que permite inserir manualmente estas coordenadas ou, opcionalmente, adquiri-las diretamente no local, recorrendo ao recetor GPS. A informação é armazenada num ficheiro no formato *.txt* e posteriormente passada ao bloco ‘Reprodução’.

- **Reprodução**

Este bloco, assegurado por uma aplicação em ambiente Windows, é o elemento central do projeto. Recolhe todos os dados provenientes do ficheiro de rota, do sensor inercial e do recetor GPS, para, em função deles, definir em que instantes deve ser iniciada a reprodução, escolher a trilha adequada e processá-la para permitir audição binaural de acordo com o contexto em que se encontra o ouvinte.

Internamente, pode dividir-se em outros dois blocos funcionais, como mostra a Figura 1-11:



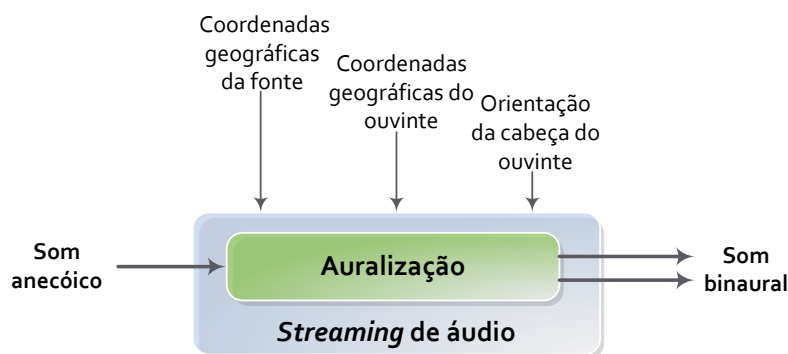
**Figura 1-11:** Composição do bloco de reprodução

- **Controlador de reprodução**

Este bloco cruza os dados lidos pelo recetor GPS com os dados provenientes do ficheiro de rota, para decidir se o bloco ‘Auralização’ deve ou não ser ativado.

- **Auralização**

Neste bloco (vide Figura 1-12) reside o foco principal de processamento de sinal. Em função dos dados provenientes do recetor GPS, do sensor inercial e do bloco ‘Controlador de reprodução’ (posição da fonte sonora virtual), o som anecoico é processado para criar o efeito de áudio 3D (auralização) em tempo real. O *streaming* é ativado sempre que o bloco ‘Controlador de reprodução’ o determine.



**Figura 1-12:** Auralização e *streaming* de áudio

## 1.4 Estrutura da dissertação

Tendo em conta a constituição do sistema ilustrada na Figura 1-10, esta dissertação foi dividida em seis capítulos.

**Capítulo 1 (Introdução)** – O presente capítulo procura explicar a motivação que levou a este projeto, apresenta sucintamente o estado de arte para sistemas de *audio-guiding* e, por fim, mostra, numa perspetiva geral, o sistema implementado neste trabalho.

**Capítulo 2 (Sistema de auralização)** – Apresenta uma abordagem teórica ao processo de auralização e explica como é implementado o *streaming* de áudio. Expõe em seguida a implementação de duas aplicações que executam o processo de auralização e de *streaming* em conjunto.

**Capítulo 3 (Sistemas de deteção de posição e orientação)** – Começa por enquadrar os dispositivos de aquisição de dados (sensor inercial e recetor GPS) no contexto deste trabalho. Após isso, apresenta os dispositivos usados e as suas principais características (elétricas, comunicação e sistemas de coordenadas). No fim, explica a implementação de duas aplicações que realizam a aquisição de dados com estes dispositivos.

**Capítulo 4 (Integração dos sistemas)** – Explica a implementação e interfaces de duas aplicações ('Gestor de rota' e 'Sistema de Reprodução') que integram os vários sistemas desenvolvidos nos capítulos anteriores.

**Capítulo 5 (Testes e validação)** – Expõe um conjunto de testes subjetivos realizado com o protótipo desenvolvido e procede à interpretação e discussão dos resultados obtidos.

**Capítulo 6 (Conclusões e trabalho futuro)** – Neste último capítulo são globalmente discutidos os resultados e extraídas conclusões desta dissertação. São também deixadas algumas diretrizes para trabalho futuro.



## 2. Sistema de Auralização

Este capítulo faz uma descrição do procedimento de auralização, da biblioteca *Auralisation Models and Applications for Virtual Reality Environments* (AcousticAVE) e do desenvolvimento de duas aplicações, escritas em linguagem C, que farão a auralização e o *streaming* de áudio. Uma destas será o alicerce à implementação do bloco ‘Auralização’ no contexto da aplicação final (bloco ‘Reprodução’), Figura 1-10.

### 2.1 Introdução

#### 2.1.1 Auralização

O termo auralização apareceu como uma analogia à visualização, mas aqui para descrever a audição de campos sonoros virtuais.

O processo de auralização visa a recriação do campo sonoro criado por fonte(s) num dado espaço, de forma a transmitir a sensação de que o ouvinte se encontra dentro desse espaço, numa certa posição [9]. Por outras palavras, consiste em produzir uma experiência auditiva binaural realista a partir de dados numéricos (medidos, simulados ou sintetizados) [10].

É necessário ter em conta todos os aspetos influentes na sensação de proveniência do som: posições do ouvinte e da fonte sonora e características geométricas e físicas (materiais) do espaço modelado, incluindo as relacionadas com a presença do ouvinte, nomeadamente ouvidos, cabeça e torso.

Podemos considerar que esta combinação de aspetos é um sistema. Tendo em conta as leis que regem a propagação do som, esse sistema pode ser considerado sistema linear e invariante no tempo (*Linear Time-Invariant* – LTI) [10].

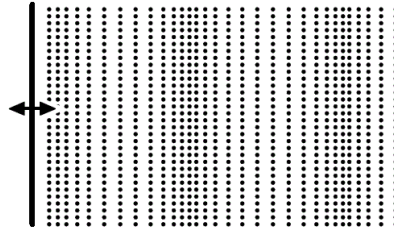
Assim, a auralização pode ser conseguida através da convolução de um som anecóico com a resposta impulsional de uma sala (*Room Impulse Response* – RIR). Também podemos reconstituir uma experiência auditiva que reflita a influência da anatomia humana (ouvidos, cabeça e torso) através da convolução de um som anecóico com a *Head-Related Impulse Response* (HRIR).

#### 2.1.2 Som

O som resulta da propagação de uma onda (frente de compressão) mecânica, num meio gasoso, líquido ou sólido. As ondas sonoras têm origem em objetos em vibração, que por sua vez transmitem essa vibração

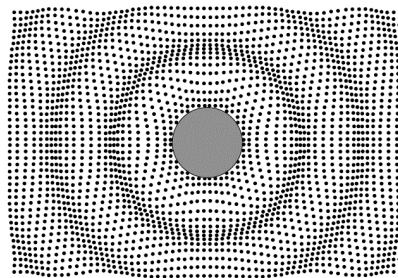
(aos átomos) ao longo do percurso de propagação até aos nossos tímpanos. O sistema auditivo interno forma, a partir dos estímulos recebidos nos tímpanos, a percepção do som. Desde que a frequência dessa vibração se encontre compreendida aproximadamente entre 20Hz e 20KHz [10].

O som no ar propaga-se sob a forma de ondas longitudinais (compressões e rarefações). A Figura 2-1 ilustra uma onda num contexto unidimensional, com a fonte sonora no lado esquerdo.



**Figura 2-1:** Onda longitudinal [10]

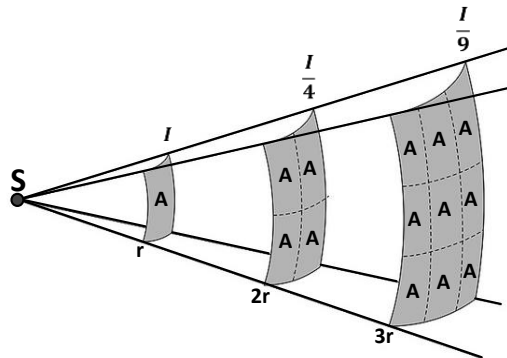
Idealmente, no ar ou outro meio gasoso isotrópico, quando o meio é estimulado com uma vibração originada num ponto, essa vibração propaga-se esfericamente [10]. A Figura 2-2 apresenta um corte 2D para ilustrar essa situação.



**Figura 2-2:** Radiação com origem em um ponto (ilustração 2D) [10]



A energia dessa vibração vai-se espalhando uniformemente pela superfície dessa frente de onda esférica, enquanto essa frente se afasta do seu centro, ou seja, da fonte sonora ( $S$ ).



**Figura 2-3:** Frente de onda esférica em relação ao seu raio

Depreende-se que a intensidade média da onda ( $I$ ) diminui com o do quadrado da distância.

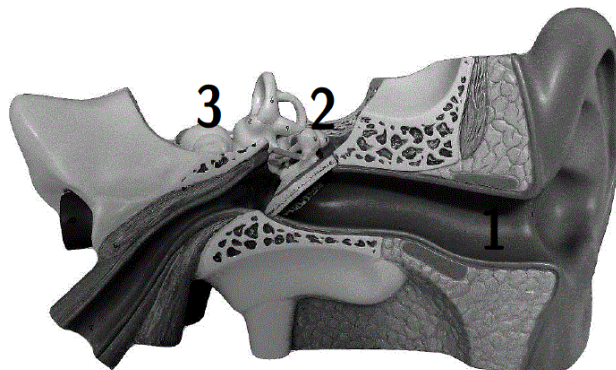
$$I = \frac{P_{source}}{4\pi r^2} \quad (2-1)$$

Na equação,  $P_{source}$  é a potência emitida pela fonte e  $r$  é o raio da frente de onda esférica.

É de notar que quando o meio não é isotrópico, ou mesmo com obstáculos (dependendo da geometria e superfícies da sala), efeitos como a difração, atenuação e reflexão podem acontecer. E em caso de movimento da fonte em relação ao ouvinte, o ouvinte terá a percepção de mudanças de frequência do som emitido pela fonte (efeito de *Doppler*).

### 2.1.3 Anatomia do ouvido humano

Conhecer o ouvido humano e o seu processamento de sinal correspondente é importante para perceber em que medida este órgão sensorial afeta a nossa percepção espacial do som. O nosso sistema de audição periférica está dividido em 3 partes, nomeadamente: ouvido externo (1), ouvido médio (2) e ouvido interno (3), Figura 2-4.

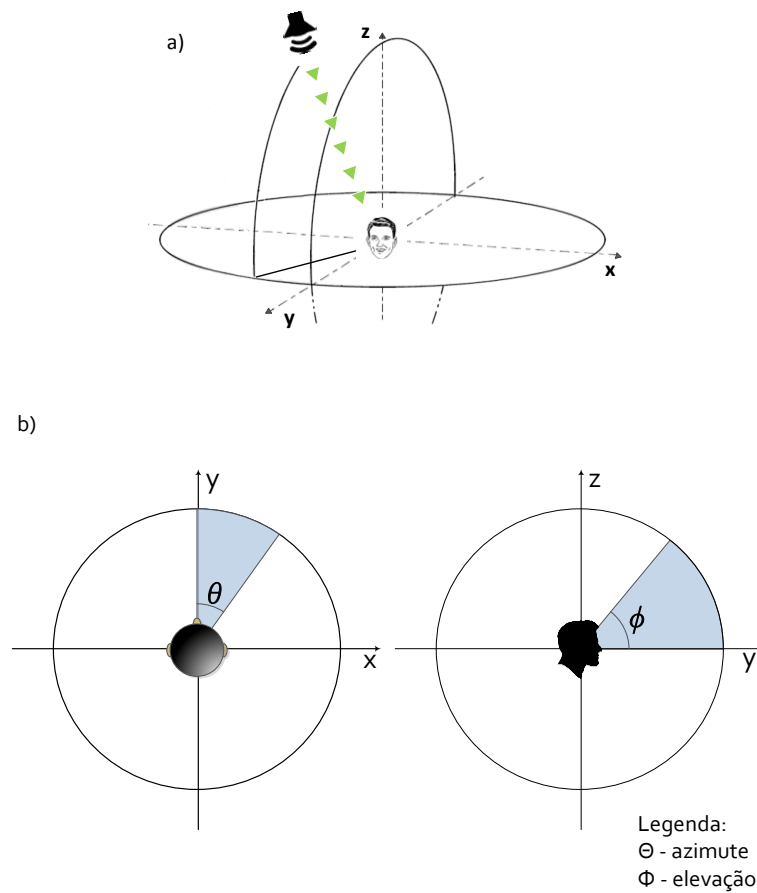


**Figura 2-4:** Anatomia do ouvido humano [10]

O ouvido externo é composto por duas partes, o pavilhão auditivo, comumente conhecido por orelha, e o canal auditivo. Podemos ver que o pavilhão auditivo é assimétrico o que permite gerar pistas espectrais praticamente únicas para cada ângulo de incidência. Pistas essas formadas por fenômenos de reflexão e refração no pavilhão auditivo conduzem o som até aos nossos canais auditivos (e assim afetam a nossa percepção do som). O pavilhão auditivo também responsável por amplificar o som injetado no canal auditivo, ajudado assim a termos uma melhor percepção do som [9]. O canal auditivo é aproximadamente um tubo que termina no tímpano. Tornando este canal um oscilador  $\lambda/4$  com uma frequência fundamental de ressonância de aproximadamente de 3KHz, colocando nesta faixa a máxima sensibilidade auditiva [10], que é só a frequência média usada no discurso humano [10]. Contudo para uma audição de sinais com altas frequências o canal auditivo já não tem influência. Depois de combinando todos os efeitos da cabeça, torso e pescoço, pavilhão auditivo e canal auditivo o som chega ao tímpano. Posteriormente, o som é propagado mecanicamente do ouvido médio até o ouvido interno, com bastante ganho também [9], onde este, por fim, transmite a informação sonora captada ao cérebro através de impulsos nervosos.

#### 2.1.4 Audição binaural

A audição binaural permite detetar a proveniência de som. Para isso terão que existir duas respostas, uma para o ouvido direito e outra para o ouvido esquerdo. Estas respostas são as HRIR. Elas refletem os efeitos que a cabeça, tronco, ombros e ouvidos terão sobre o som que será ouvido e proporcionam a percepção da direção de proveniência do som, principalmente através das diferenças temporais e de intensidade do som que chega aos ouvidos (devidas a distorções lineares que são diferentes para cada ouvido), respetivamente *Interaural Time Difference* (ITD) e *Interaural Intensity Difference* (IID) [10]. As transformadas de *Fourier* das HRIR designam-se HRTF e são usadas como filtros. Para cada direção existem duas HRTF, uma para o ouvido direito e outra para o esquerdo, e a referenciação no espaço de cada par de HRTF é normalmente feita através de um sistema de coordenadas *head-related* [10] que considera a cabeça do ouvinte no centro do referencial. Para cada par azimute/elevação (direção) existe um par de HRTF, como mostra a Figura 2-6. O movimento da cabeça é pois um fator muito importante para a nossa deteção da proveniência do som.

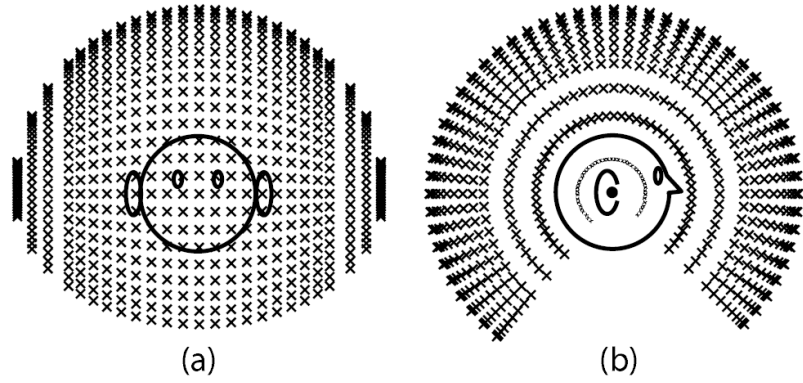


**Figura 2-6:** a) Sistema de coordenadas; b) Projeções dos ângulos de azimuth e elevação

As HRTF tipicamente são medidas, em câmeras anecóicas, com recurso a manequins equipados com microfones nos 'ouvidos' e à emissão de impulsos ao longo de uma grelha finita de amostragem espacial com formato esférico, centrada na cabeça do manequim. As respostas aos impulsos captados pelos microfones constituem as HRIR para a respetiva direção, como mostra as Figura 2-5 e Figura 2-7.



**Figura 2-5:** Manequim de aquisição de HRTF (fonte: <https://www.ece.ucdavis.edu/cipic/spatial-sound/research/>)

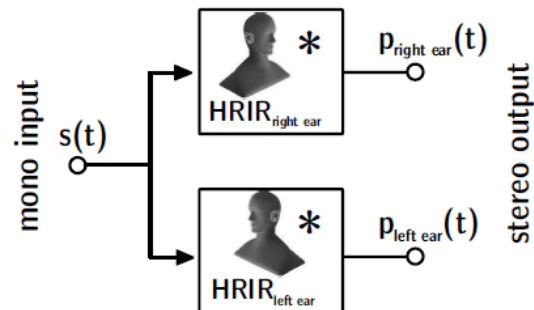


**Figura 2-7:** Esquema de localização dos pontos de emissão na medição de HRTF a) vista frontal b) vista lateral (fonte: [https://www.ece.ucdavis.edu/cipic/files/2015/04/cipic\\_CIPIC\\_HRTF\\_Database.pdf](https://www.ece.ucdavis.edu/cipic/files/2015/04/cipic_CIPIC_HRTF_Database.pdf))

Recorrendo a uma base de dados de HRTF, é possível aplicar pistas de espacialização 3D apropriadas a qualquer gravação anecóica e assim conseguir uma experiência de auralização binaural. Basta realizar a operação de convolução do sinal áudio monofónico com o par de HRIR de uma dada direção. As equações seguintes descrevem o processo, sendo  $s(t)$  o som monofónico e  $p(t)$  os sinais sintetizados para cada ouvido [10] – vide Figura 2-8.

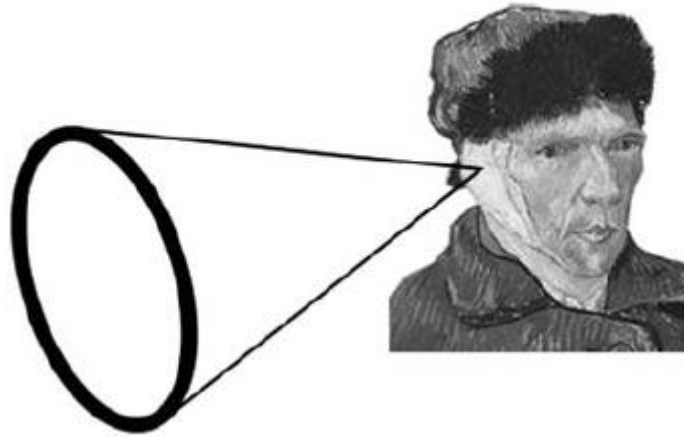
$$p_{\text{ouvido direito}}(t) = s(t) * \text{HRIR}_{\text{ouvido direito}} \quad (2-2)$$

$$p_{\text{ouvido esquerdo}}(t) = s(t) * \text{HRIR}_{\text{ouvido esquerdo}} \quad (2-3)$$



**Figura 2-8:** Processo de auralização binaural [10]

Duma forma geral, os mamíferos são capazes de localizar uma fonte sonora usando as pistas de ITD e IID, já referidas. No entanto, em qualquer cone cujo eixo coincida com o eixo interaural, essas pistas não variam ao longo da circunferência que delimita a base [11], como ilustra a Figura 2-9.



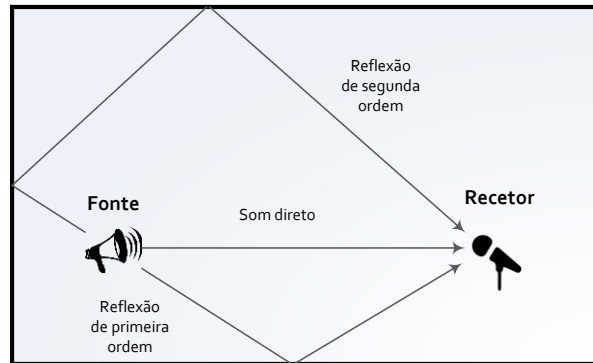
**Figura 2-9:** Cone de confusão [9]

Consequentemente, existe ambiguidade na determinação da origem das ondas sonoras que se propagam sobre a superfície do cone, independentemente da distância da fonte sonora – daí o nome ‘cone de confusão’. Evidentemente, a ambiguidade é de menor importância para fontes sonoras muito próximas. Em qualquer caso – e este é o aspecto mais importante – pode sempre ser resolvida através de movimentos da cabeça. Alguns mamíferos usam inclusivamente a capacidade de movimentar as orelhas [11].

### **2.1.5 Percepção acústica em salas**

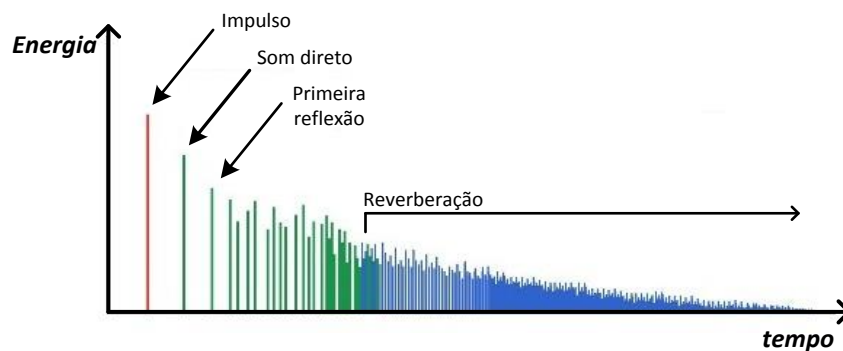
Quando uma fonte de som emite numa sala, as ondas sonoras, no seu trajeto até ao recetor, são refletidas nos elementos construtivos da sala, difratadas em esquinas e fendas, espalhadas pelas texturas das superfícies, parcialmente absorvidas pelos materiais das superfícies, atenuadas por absorção no ar e sofrem interferência entre si. Assim, a percepção auditiva humana depende não só da excitação da fonte, mas também da influência dos fenómenos descritos. Se o sinal da fonte for um único impulso que excita todas as frequências de interesse, e o recetor monofónico for, por exemplo, um microfone, esta resposta será a RIR, que pode ser vista como a "impressão digital acústica" da sala [12].

Supondo que o percurso do som pode ser representado por raios (i.e. linhas retas como ilustra a Figura 2-10) a resposta impulsional da sala abarca informação, não só sobre a direção e intensidade de frente de onda direta (no caso de haver visibilidade entre a fonte e o recetor), como também sobre a direção e intensidade de chegada de cada reflexão.



**Figura 2-10:** Medição da RIR em uma sala

Ao considerar-se um período de tempo adequado e/ou ordem de reflexão desejada, podemos obter respostas como a ilustrada na Figura 2-11:



**Figura 2-11:** Resposta impulsional de uma sala genérica

Tendo em conta as propriedades das respostas e também a percepção humana podemos dividir estas respostas em 3 partes: som direto, reflexões iniciais e reverberação (ou reflexões de ordem superior) [12].

- **Som direto**

É o primeiro e mais forte impulso a chegar ao recetor e é decisivo para a localização da fonte sonora (efeito de precedência). O som direto sofre um atraso proporcional à distância entre fonte e receptor e é atenuado apenas pela absorção no ar (no caso de haver visibilidade direta).

- **Reflexões iniciais**

São as primeiras reflexões que chegam ao ouvinte e têm origem na superfícies circundantes da sala ou em obstáculos que existam na sala. Estas reflexões são mais atrasadas e atenuadas devido a caminhos mais longos na propagação do som e consequentemente perdem energia com as absorções

encontradas nas trajetórias descritas. O efeito perceptual destas reflexões é reforçar o som direto; a audição humana não consegue identificar separadamente (até um intervalo máximo de 20ms, a não ser que o intervalo entre elas exceda 20ms, caso em que são ouvidas como ecos), mesmo que seu nível seja até 10dB superior em relação ao som direto (efeito de *Haas*). Destas primeiras reflexões podem ser tiradas informações sobre a fonte sonora: posição, distância, largura da fonte e intensidade [12].

- **Reverberação (ou reflexões de ordem superior)**

Encontram-se normalmente com um atraso de 50-80ms em relação do som direto. A partir desse tempo de atraso o número de reflexões aumenta gradualmente e a percepção auditiva humana não consegue perceber como sendo eventos únicos, formando um campo sonoro difuso, ou seja, uma reverberação, que é quase independente da posição do ouvinte [12]. A reverberação é possivelmente o atributo acústico mais distinto de uma sala, uma vez que certas particularidades como o volume e a forma da sala, estão diretamente associados à reverberação, dando à sala um som característico e quase particular. Estas reflexões de ordem superior em última instância criam uma sensação de envolvimento no ouvinte [10].

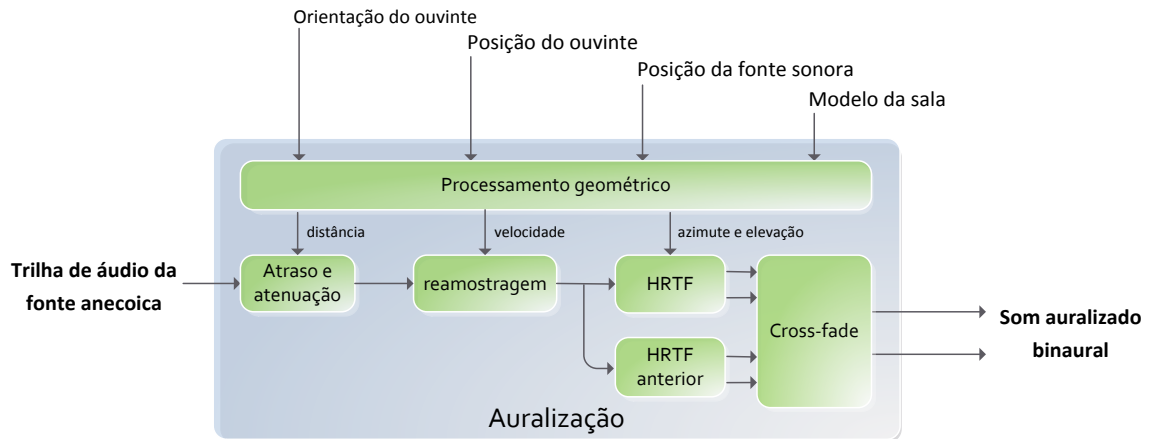
Uma vez que o protótipo desenvolvido neste trabalho tem como propósito operar em ambiente exterior, teve-se em conta apenas o som direto (desprezando a única reflexão que poderia ser relevante, que é uma reflexão de primeira ordem no chão entre a fonte e o ouvinte). Considerar as reflexões e a reverberação poderá ser importante numa eventual expansão do sistema para utilização em ambientes interiores.

### 2.1.6 AcousticAVE

Para implementar o efeito de espacialização 3D pretendido no projeto foi usada a biblioteca AcousticAVE (LibAAVE) [13]. Esta biblioteca, escrita na linguagem C (vide Anexo B) é um motor de auralização em tempo real desenhado para ambientes interiores e reprodução binaural (i.e. para auscultadores). Contém a base de dados de HRTF usada para este projeto. Para reproduzir o campo sonoro 3D de um ambiente virtual, os dados necessários são:

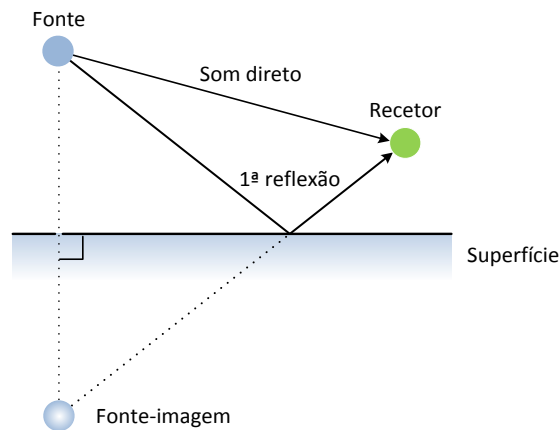
- Posição(ões) da(s) fonte(s) sonora(s);
- Posição e orientação da cabeça do ouvinte;
- Modelo de configuração física da sala (forma, dimensões, materiais constituintes)
- Trilha áudio mono (gravação anecóica) para cada fonte sonora.

A figura seguinte ilustra a interface e os blocos funcionais do processo de auralização.



**Figura 2-12:** AcousticAVE: processo de auralização

Esta biblioteca suporta mudança de posição das fontes sonoras, do ouvinte ou ambas em simultâneo [13]. A influência da sala é simulada por um modelo computacional geométrico da propagação de som (método das imagens virtuais ou *Mirror-Image Source* – MIS) [9]. Neste modelo, as ondas sonoras que provêm da(s) fonte(s) anecóica(s) são tratadas como raios, que sofrem reflexão e atenuação nas superfícies da sala. São determinados o azimute e a elevação de cada raio sonoro em relação ao ouvinte. As reflexões são vistas como tendo origem numa fonte virtual obtida por espelhamento da fonte original na superfície de reflexão (*mirror-image source*), como ilustra a Figura 2-13. Para o cálculo de múltiplas reflexões, este processo é aplicado recursivamente [14]. Quanto à reverberação, a biblioteca AcousticAVE implementa um algoritmo simples de que adiciona uma cauda de reflexões tardias à saída [14].



**Figura 2-13:** Fonte sonora e sua imagem virtual definindo a direção de uma reflexão de primeira ordem

O modelo computacional permite calcular a atenuação e o atraso de cada onda sonora no trajeto até ao ouvinte (bloco ‘Atenuação e atraso’) e simular o efeito de *Doppler*, que ocorre quando existe movimento do ouvinte ou de fontes (bloco ‘Reamostragem’); as várias componentes de áudio binaural são construídas aplicando as HRTF adequadas a cada onda (convolução pelo algoritmo *overlap-add*), selecionadas tendo em



conta azimute e elevação relativamente ao ouvinte. As descontinuidades entre trechos de áudio processados com HRTF diferentes (que provocariam artefactos audíveis) são eliminadas por *cross-fade* [14].

A biblioteca permite escolher entre quatro conjuntos de HRTF, obtidos em bases de dados de domínio público: MIT KEMAR HRTF [15], CIPIC [16], TU-Berlin HRTF [17] e LISTEN HRTF Database [18].

Como neste trabalho só se pretende que seja reproduzida uma fonte sonora virtual (fonte primária) de cada vez e em ambiente exterior (portanto sem necessidade de considerar modelo da sala, pois não existem reflexões ou reverberação), a operação em tempo real pode ser facilmente alcançada, não exigindo assim um enorme esforço computacional da plataforma em que seja executada.

## 2.2 Aplicações

Este trabalho envolve o desenvolvimento de aplicações em que se pretende funcionamento autónomo (*standalone*) exigindo interfaces gráficas e recurso a bibliotecas de apoio. Por isso, foram avaliadas várias ferramentas de desenvolvimento de aplicações em C/C++, entre as quais *Eclipse* [19], *Microsoft Visual Studio* [20] e *Qt Creator* [21] (vide Anexo A). Escolheu-se esta última, uma vez que é *cross-platform*, facilitando a portabilidade das aplicações criadas, integra um vasto leque de bibliotecas e o seu *Integrated Development Environment* (IDE) apresenta-se como o que mais facilita a implementação de interfaces gráficas (GUI – *Graphical User Interface*).

Nesta secção, são explicadas duas aplicações desenvolvidas tendo em vista o bloco ‘Auralização’ e o *streaming* de áudio (vide Figura 1-12).

Como ponto de partida, recorreu-se a um exemplo (*circle.c*) apresentado na documentação da biblioteca AcousticAVE. Trata-se de um caso em que o ouvinte se mantém parado no centro de um referencial cartesiano, com a cabeça orientada no sentido positivo do eixo *yy*, e o eixo interaural coincidente com o eixo dos *xx*. A fonte sonora descreve circunferências à sua volta no plano *xy*, sempre à mesma distância (5 metros), enquanto emite uma senoide a 1000Hz.

Considerou-se mais interessante inverter a situação, colocando a fonte sonora parada e o ouvinte a descrever circunferências em redor dela, uma vez que, nas aplicações previstas, serão as fontes sonoras (loais de atração turística) e não os ouvintes a deslocar-se.

A AcousticAVE requer algumas configurações iniciais, relativas às posições da(s) fonte(s) e do ouvinte, ao modelo geométrico e parâmetros acústicos da sala a considerar, à escolha do conjunto de HRTF (uma vez que contém quatro), à ordem de reflexão máxima a considerar e à aplicação ou não de reverberação.

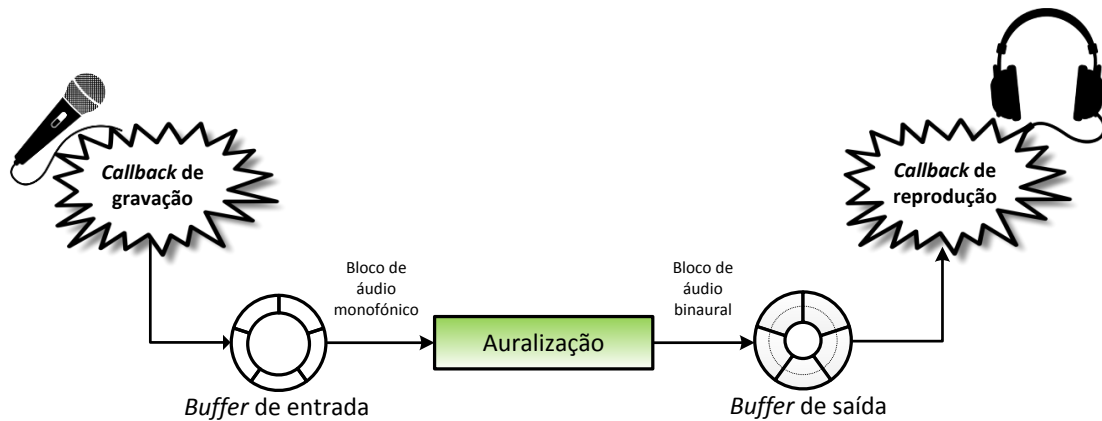
Como se pretendia auralização em espaço exterior, a biblioteca foi configurada para não considerar reflexões nem reverberação. Nestas condições, é irrelevante a escolha do modelo acústico da sala. Foi aplicado o conjunto de HRTF “MIT KEMAR”.

Para realizar o *streaming* de áudio (Figura 1-12) utilizou-se a biblioteca PortAudio [22] (vide Anexo B), pois permite garantir baixa latência. Isto é essencial para que a experiência auditiva seja convincente, ou seja, a perceção dos efeitos sonoros de alterações da posição e/ou orientação da cabeça do ouvinte seja praticamente instantânea. A PortAudio permite dois modos de *streaming*: síncrono e assíncrono [23]. O modo síncrono é suficiente quando podem ser toleradas latências relativamente elevadas. Porém, sempre que se exijam baixas latências (como é o caso das duas aplicações descritas na presente secção) é necessário recorrer ao modo assíncrono. Este usa funções de *callback* desencadeadas por interrupções de alta prioridade sempre que é necessário reproduzir ou adquirir áudio (i.e. preencher o *buffer* interno de saída ou recolher dados do *buffer* interno de entrada) [23].

A primeira aplicação, descrita em 2.2.1, irá efetuar o processamento de auralização descrito utilizando como fonte virtual o sinal adquirido por um microfone; a segunda aplicação, descrita em 2.2.2, irá efetuar o mesmo processamento de auralização utilizando como fonte virtual o sinal proveniente de um ficheiro, gerando ambas áudio binaural (dois canais: ‘L’ e ‘R’).

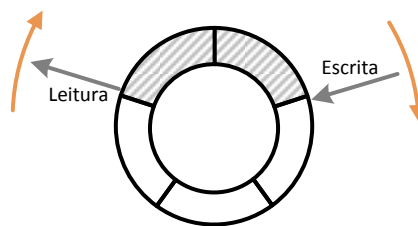
### 2.2.1 Auralização com *streaming* a partir de microfone

Foram implementados dois *buffers* circulares, um entre a *callback* de gravação e o bloco de auralização e outro do bloco de auralização para a *callback* de reprodução, como mostra a Figura 2-14. O bloco de auralização, assegurado pela biblioteca AcousticAVE, foi configurado com o exemplo referido na secção anterior (*circle.c*).



**Figura 2-14:** Primeira aplicação de ensaio do sistema de auralização

Cada *buffer* tem dois ponteiros associados, um de leitura e outro de escrita. Como mostra a Figura 2-15, o primeiro aponta sempre a primeira posição já preenchida com dados para ler (blocos a cinza) e o segundo aponta sempre a primeira posição livre para escrever novos dados (blocos a branco). Os ponteiros são incrementados no sentido indicado.



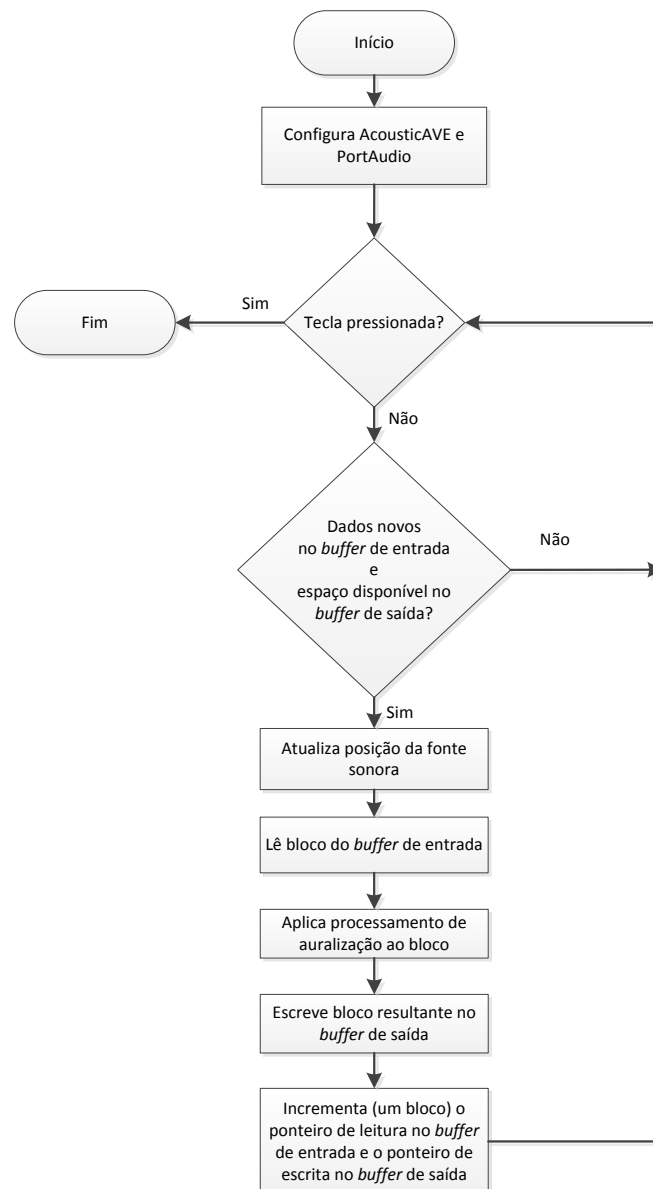
**Figura 2-15:** Buffer circular e posicionamento dos ponteiros de escrita e leitura

A *callback* de gravação é responsável por incrementar o ponteiro de escrita no *buffer* de entrada, pois é ela que insere amostras áudio nesse *buffer*. De um modo análogo, a *callback* de reprodução é responsável por incrementar o ponteiro de leitura do *buffer* de saída, pois é ela que retira as amostras áudio desse *buffer*. O bloco de auralização controla o ponteiro de leitura do *buffer* de entrada e o ponteiro de escrita do *buffer* de saída, pois é ele o responsável por retirar e inserir amostras áudio nos *buffers* de entrada e saída, respetivamente.

O processamento de auralização (função *main* da aplicação) é realizado bloco a bloco, como se mostra na Figura 2-16. O número de amostras áudio por bloco (*FRAMES\_PER\_BLOCK*) é predefinido; neste caso fez-se *FRAMES\_PER\_BLOCK*=1024, o que, a uma frequência de amostragem de 44100Hz (*SAMPLE\_RATE*=44100) corresponde a aproximadamente 23.2ms de áudio. Usaram-se amostras de 16bits

(*SAMPLE\_TYPE=2*). Como o processamento de auralização é binaural, são geradas duas amostras (canais esquerdo e direito) por cada amostra anecóica do sinal de entrada. Assim, o espaço de memória ocupado pelos blocos do *buffer* de saída é o dobro do ocupado pelos do *buffer* de entrada.

É desejável minimizar a latência, para que as mudanças de posição e orientação do ouvinte sejam refletidas auditivamente com o mínimo atraso possível (idealmente imperceptível). Este objetivo é favorecido por *buffers* (quer de saída quer de entrada) com pequeno número de blocos. No entanto, um número excessivamente pequeno põe em causa o funcionamento ininterrupto da gravação/reprodução. Neste caso, os *buffers* foram configurados para conter 5 blocos (*IBUF\_BLOCKS=5* e *OBUF\_BLOCKS=5*), o que se revelou um compromisso apropriado.

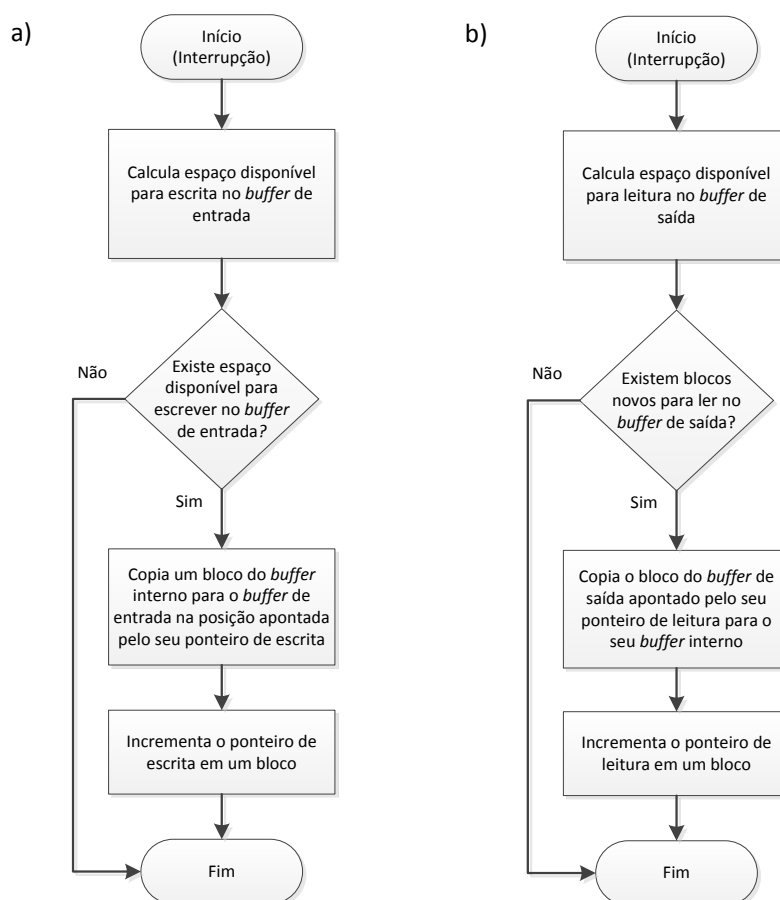


**Figura 2-16:** Fluxograma da primeira aplicação de ensaio

Como já foi referido, as funções de *callback* são executadas sempre que existir necessidade de aquisição ou reprodução; são portanto executadas várias vezes durante o processo de auralização, que decorre até uma tecla ser premida.

A *callback* de gravação tem como função transferir blocos de um *buffer* interno da PortAudio, que guarda as amostras adquiridas pelo microfone, para o *buffer* de entrada, como descreve a Figura 2-17 a).

Por sua vez, a *callback* de reprodução tem como função transferir blocos de áudio binaural do *buffer* de saída para outro *buffer* interno da PortAudio onde são guardadas as amostras áudio a enviar para a saída *stereo*, como descreve a Figura 2-17 b).



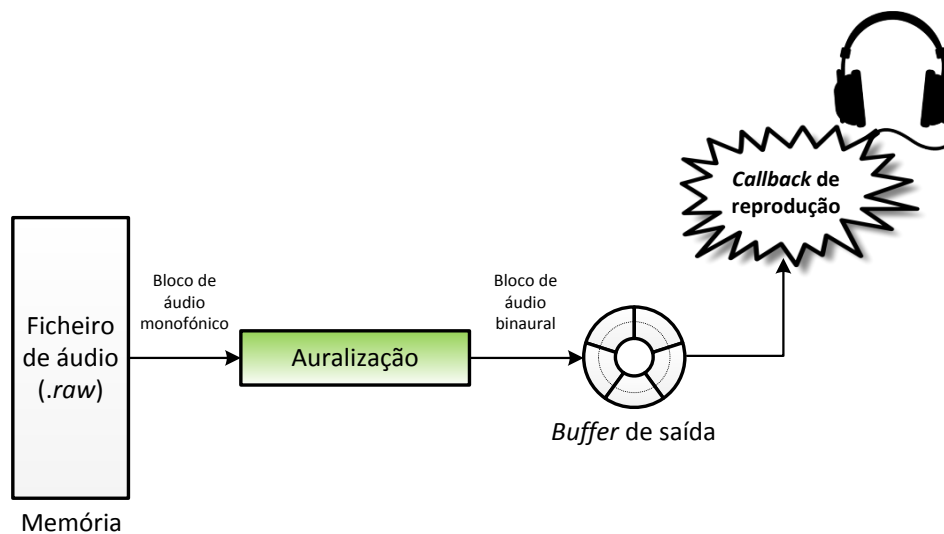
**Figura 2-17:** Fluxogramas de funcionamento das funções de *callback*: a) de gravação; b) de reprodução

Antes de escrever ou ler dados de algum *buffer*, qualquer dos elementos da aplicação verifica sempre se existe espaço disponível para escrita ou leitura. Essa verificação baseia-se no cálculo da diferença entre as posições dos ponteiros respetivos.

O código desta aplicação, designado *MicProcessingStreaming.pro*, encontra-se no CD que acompanha esta dissertação (vide Anexo C).

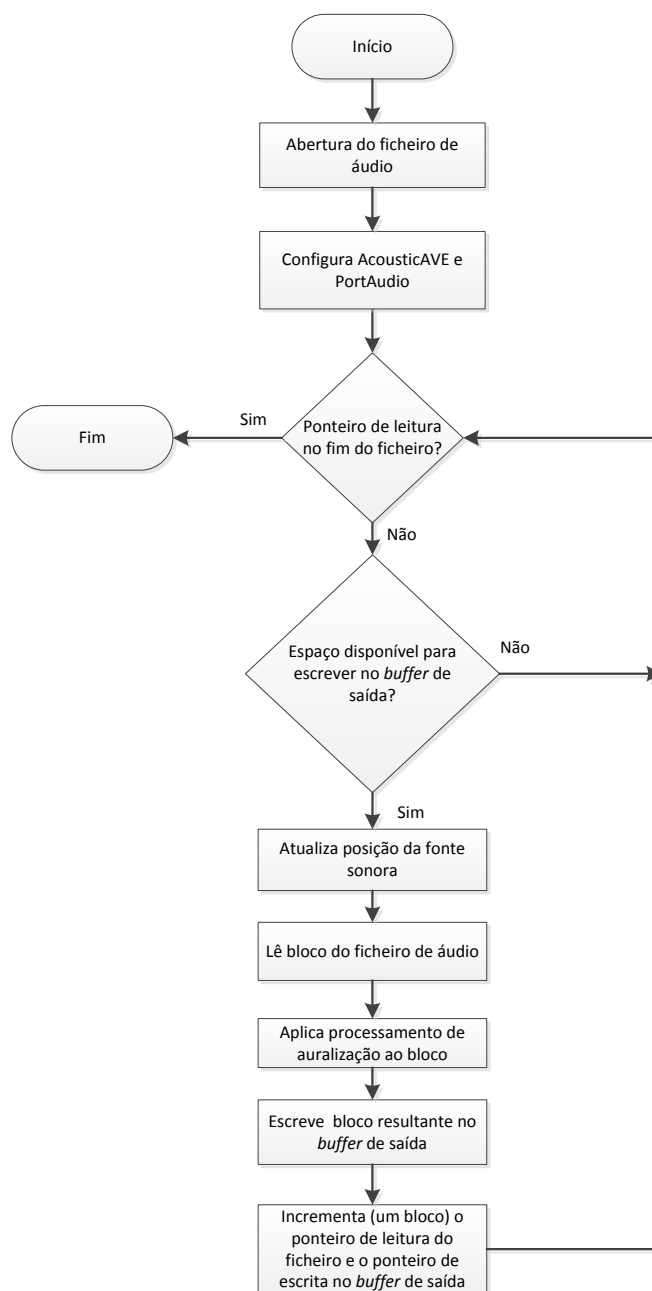
### 2.2.2 Auralização com *streaming* a partir de ficheiro áudio

Esta aplicação apresenta um funcionamento semelhante à descrita na secção 2.2.1, mas as amostras áudio provêm agora de um ficheiro de áudio monofónico e não de um microfone, como mostra a Figura 2-18. Usaram-se ficheiros em formato *.raw* (que é um formato sem *header*, ao contrário do formato *.wav*, por exemplo), sem compressão, com 16bits por amostra e uma frequência de amostragem de 44100Hz. Nesta situação, pode-se considerar que o *buffer* de entrada é o ficheiro de áudio aberto em memória, não havendo necessidade de usar uma *callback* de gravação.



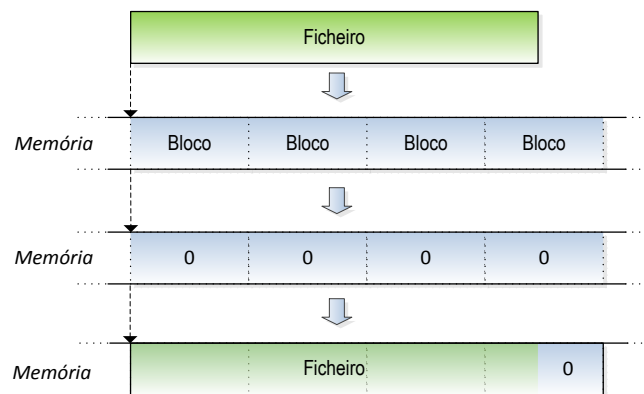
**Figura 2-18:** Segunda aplicação de ensaio do sistema de auralização

O ficheiro de áudio agora tem um ponteiro de leitura associado que é incrementado pelo processo de auralização (função *main* do programa), também em blocos de 1024 amostras. Do ponto de vista do *buffer* de saída e da *callback* de reprodução, tudo se mantém como descrito na secção anterior. É a *callback* de reprodução que impõe o ritmo de funcionamento ao sistema: sempre que ela consumir um bloco binaural do *buffer* de saída, será processado o bloco seguinte do ficheiro, resultando um novo bloco binaural no *buffer* de saída. A Figura 2-19 mostra a função *main* da aplicação. Comparando com a Figura 2-17, é de notar que agora a aplicação termina quando for completada a leitura dos dados do ficheiro áudio (previamente copiados para um espaço de memória reservado para o efeito, como se explica no parágrafo seguinte) e não quando uma tecla for premida.



**Figura 2-19:** Fluxograma da segunda aplicação de ensaio

Refiram-se os cuidados tidos na alocação de memória para o ficheiro áudio para precaver o caso geral, em que o tamanho do ficheiro não será múltiplo inteiro de *FRAMES\_PER\_BLOCK*. Como o espaço reservado em memória corresponde sempre a um número inteiro de blocos, este número é calculado arredondando por excesso o quociente entre o número total de amostras do ficheiro e o número de amostras por bloco (*FRAMES\_PER\_BLOCK*) e a parte sobranete do último bloco é preenchida com silêncio (amostras de amplitude 0), como ilustra a Figura 2-20.



**Figura 2-20:** Alocação e inicialização de memória para o áudio de entrada

O código desta aplicação, designado *FileProcessingStreaming.pro*, encontra-se no CD que acompanha esta dissertação (vide Anexo C).





## 3. Sistemas de detecção de posição e orientação

Neste capítulo, são estudados os dois sistemas que irão fornecer aos sistemas ‘Controlador de reprodução’ e ‘Auralização’ a posição e orientação da cabeça do ouvinte, de modo ajudar o funcionamento destes blocos para cada cenário de posição e orientação da cabeça. Também são desenvolvidas 2 aplicações mas em C++ com estes sistemas de modo a testa-los para fazer parte integrante do bloco de ‘Reprodução’ e do bloco de ‘Gestor de rota’ (no caso do sistema de posição).

### 3.1 Detecção de posição

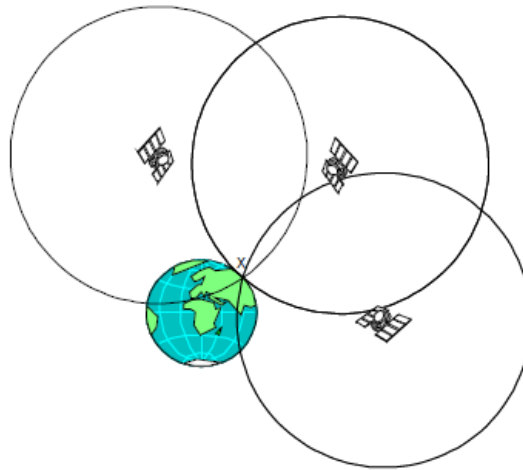
#### 3.1.1 Introdução

É necessário atualizar constantemente a posição do ouvinte, com dois objetivos:

- Calcular (tendo em conta também a orientação da cabeça) a proveniência do som emitido em termos de azimuth e elevação relativamente à cabeça do ouvinte, e assim permitir ao sistema de auralização escolher o par de HRTF adequado.
- Identificar se o ouvinte entrou numa região de *trigger* de uma fonte virtual, para iniciar a reprodução da gravação correspondente.

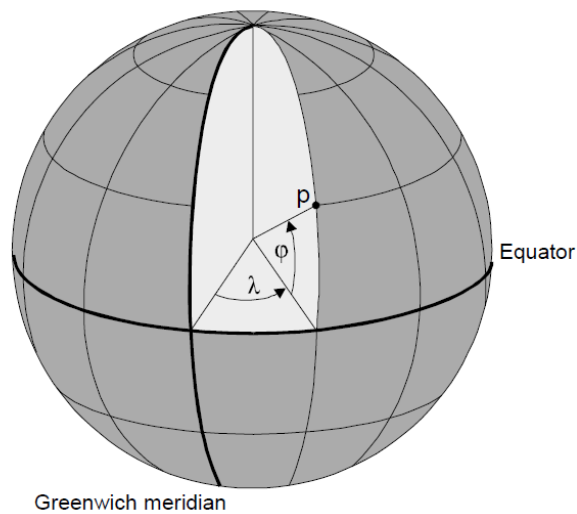
Para este efeito, usou-se um recetor GPS.

O sistema de GPS foi criado em 1973 para fins militares, mas hoje em dia o seu uso civil também é permitido, sendo administrado pelo Departamento de Defesa dos Estados Unidos [24]. Os recetores GPS processam sinal proveniente de vários satélites para determinar a sua própria posição (latitude, longitude e altura) na Terra. Atualmente, o sistema de GPS opera com cerca de 31 satélites. No modo mais simples, basta haver 3 satélites no campo de “visão” do recetor, sendo a posição calculada através de uma triangulação – vide Figura 3-1. Ora, em campo aberto estão sempre visíveis pelo menos 4 satélites em virtude das suas posições em órbita [25]. A disponibilidade de mais satélites visíveis aumenta a resolução da medida de posição.



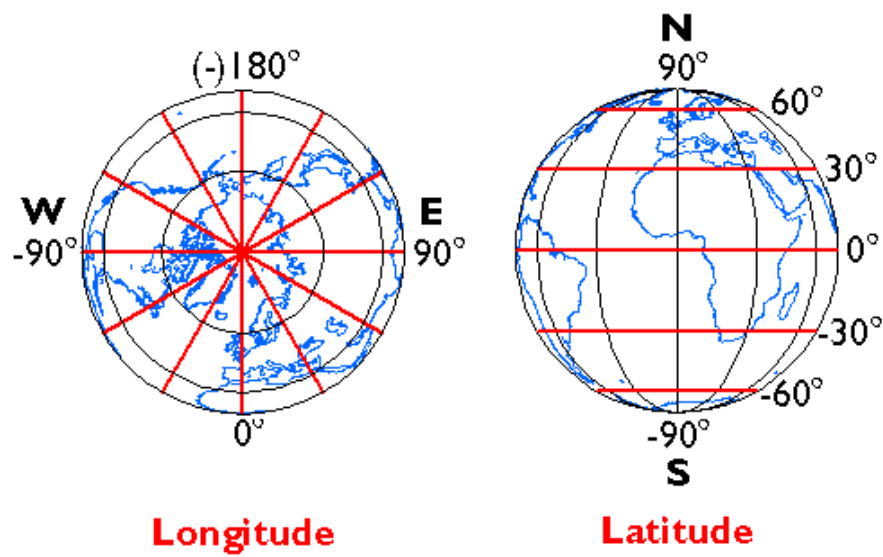
**Figura 3-1:** Triangulação com satélites [25]

As coordenadas geográficas fornecidas por recetores deste tipo baseiam-se num referencial com origem no centro do planeta Terra. Duas delas coincidem com coordenadas esféricas. Trata-se dos ângulos de *latitude* ( $\phi$ ), medido a partir do plano de Equador, e *longitude* ( $\lambda$ ), medido a partir do meridiano de *Greenwich* – vide Figura 3-2.



**Figura 3-2:** Sistema de coordenadas geográficas: ângulos de longitude ( $\lambda$ ) e latitude ( $\phi$ ) [25]

O ângulo de latitude apresenta uma variação de 0 a 90° para Norte e 0 a -90° para Sul enquanto o ângulo de longitude apresenta uma variação de 0 a 180° para Este do meridiano de *Greenwich* e de 0° a -180° para Oeste do mesmo, como mostra a Figura 3-3.



**Figura 3-3:** Convenções na medida dos ângulos de longitude e latitude (fonte: <https://www.e-education.psu.edu/>)

A terceira coordenada geográfica, a *altura* ( $h$ ), é a diferença entre a coordenada esférica  $p$  (distância ao centro – vide Figura 3-2) e o raio terrestre. Tipicamente, por razões geométricas, a precisão dos receptores GPS na determinação do posicionamento vertical (altura) é cerca de 2 a 3 vezes inferior à conseguida no posicionamento horizontal (pares longitude/latitude) [26].

### 3.1.2 Recetor GPS

Procedeu-se à análise comparativa de um conjunto de recetores GPS disponíveis no mercado, com base numa pesquisa na *Internet*. A tabela seguinte resume os resultados dessa pesquisa, salientando as características mais consideradas mais importantes para o projeto.

**Tabela 3-1:** Recetores GPS e suas características principais

Marca	Modelo	Loja	Preço	Sensibilidade	Precisão
Global Sat	ND-105C	Expansys	33,99 €	-165 dBm	< 3 m
Global Sat	BU-353-S4	Expansys	44,99 €	-163 dBm	< 2.5 m
Holux	M-215	DealExtreme	35,81 €	-159 dBm	< 2.5 m
Xucaí	GD75	DealExtreme	25,22 €	-159 dBm	< 5 m
Navilock	NL-601US	Amazon.es	45,82 €	-162 dBm	< 2.5 m

Foi escolhido o recetor Xucai GD75 (Figura 3-3), atendendo sobretudo ao seu preço competitivo. A Tabela 3-2 apresenta algumas características adicionais deste recetor.



**Figura 3-4:** Recetor GPS Xucai GD75 (fonte: <https://pt.aliexpress.com/>)

**Tabela 3-2:** Características do recetor de GPS

Interface	USB
Protocolo de comunicação	NMEA 0183 (V3.0)
Frequência máxima de refrescamento	1Hz
Tempo de arranque a frio	<33s
Temperatura de funcionamento	-10° C a 70° C

O recetor apresenta conectividade USB assente na norma NMEA (*National Marine Electronics Association*) 0183 [27], que define especificações elétricas e de formatação de informação para comunicação entre dispositivos, principalmente de uso marítimo, tais como anemómetros, giroscópios, sonares e recetores GPS [27]. Esta norma usa um protocolo simples de comunicação série assíncrona. Os dados são transmitidos em frases, ou seja, em tramas de caracteres ASCII de um *talker* para um ou múltiplos *listeners* de cada vez. Cada frase começa com o carácter "\$" e termina com os caracteres especiais <CR> <LF> (*carriage return* e *line feed*), indicando o fim de uma trama e o início de outra. Cada trama apresenta um cabeçalho composto por

cinco letras, após o carácter "\$". As duas primeiras são identificativas do *talker* (*GP* no caso de um dispositivo GPS); as três seguintes são os caracteres que identificam o tipo de frase. Segue-se uma série de campos separados por vírgulas correspondendo aos dados pertencentes a cada tipo de trama. O dispositivo GPS escolhido devolve quatro tipos de tramas. A latitude, a longitude e a altura encontram-se na mesma trama, podendo por isso as outras três tramas ser descartadas, uma vez que não contêm dados úteis ao projeto final. Estes dados encontram-se na trama iniciada por *GP*GGA (*Global Positioning System Fixed Data*)[28][29].

A trama *GP*GGA seguinte é ilustrativa e a Tabela 3-3 mostra a interpretação de cada campo.

Exemplo:

*\$GP*GGA,002153.000,3342.6618,N,11751.3858,W,1,10,1.2,27.0,M,-34.2,M,,0000\*5E

**Tabela 3-3:** Formatação da informação na mensagem *\$GP*GGA [28]

Nome do Campo	Exemplo	Unidades	Descrição
Identificador da mensagem	<i>\$GP</i> GGA		Cabeçalho
UTC Time	002153.000		hhmmss.sss (h=hour, m=minute, s= second)
Latitude	3342.6618		ddmm.mmmm (d=degree, m=minute)
Indicador N/S	N		N=north ou S=south
Longitude	11751.3858		dddmm.mmmm (d=degree, m=minute)
Indicador E/W	W		E=east ou W=west
Indicador de correção de posição	1		
Satélites usados	10		Pode variar de 0 a 12
<i>HDOP</i>	1.2		Deluição de precisão horizontal
Altitude	27.0	metros	Altitude acima do nível médio das águas do mar
Unidades	M	metros	
Separação geodésica	-34.2	metros	Diferença entre o modelo geodésico e elipsoide da Terra
Unidades	M	metros	
Idade da informação em modo diferencial	-	segundos	Campo vazio quando modo diferencial de funcionamento não usado
Identificação da estação de modo diferencial	0000		
Checksum	*5E		Código de correção de erros

Para efeitos de teste usou-se um emulador de terminal grátis chamado PuTTY [30] (vide Anexo A), com este *software* conseguimos recolher os dados em bruto e visualizar todas as tramas enviadas pelo recetor GPS. Foi configurado o *software* para uma comunicação série com os parâmetros presentes na Tabela 3-4.

Tabela 3-4: Parâmetros da comunicação série do recetor GPS

<i>Baud rate</i>	4800bps
<i>Bits de dados</i>	8
<i>Paridade</i>	Não
<i>Número de stop bits</i>	1
<i>Handshake</i>	Não

A figura mostra os resultados que podem ser obtidos em ambiente exterior.

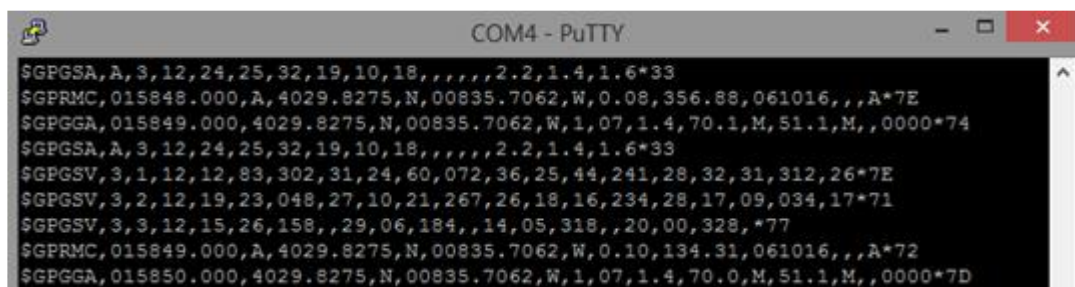


Figura 3-5: Screenshot da receção de dados do receptor GPS com PuTTY

Por exemplo, pode verificar-se que a primeira trama *\$GPGGA* indica uma latitude de 40° e 29.8275' Norte, uma longitude de 8° e 35.7062' Oeste, 7 satélites usados para a aquisição dos dados e uma altitude de 70.1 m.

### 3.1.3 Aplicação para aquisição de dados do recetor GPS

Usando o IDE *Qt Creator*, foi criada uma aplicação de teste em linguagem C++, que apresenta numa janela os dados recebidos. Para além da latitude, longitude e altitude, também se achou importante extrair da trama *\$GPGGA* o número de satélites usados, pois este constitui um indicador da precisão de medida e poderá ser útil em cenários a explorar no futuro, nomeadamente transição para situações *indoor*.

A aplicação foi assegurada por duas *threads* [31]; a primeira (que chamaremos *thread A*), encarrega-se da gestão da interface gráfica e da configuração da comunicação série do recetor; a segunda (*thread B*) responsabiliza-se pelo *parsing*<sup>1</sup> das tramas *\$GPGGA* recebidas e subsequente envio dos dados para a primeira *thread*.

A *thread B* é indispensável, porque a recolha e *parsing* das tramas constituem um evento em *loop*, como mostra a Figura 3-6. Se este *loop* fosse inserido na *thread A*, a interface gráfica bloquearia, inutilizando a aplicação [32]. A primeira tarefa da *thread B* é configurar a comunicação série entre o recetor GPS e o computador com os parâmetros contidos na Tabela 3-4, definidos pela norma NMEA 0183, que implicam receção de um carácter (8 *bits*) de cada vez. Em seguida, recolhe as tramas NMEA 0183 para uma *string* [29].

<sup>1</sup> Processo de analisar uma sequência de entrada; neste caso, identificar os caracteres da *string* que correspondem aos campos da trama *\$GPGGA*.

Quando é encontrado o carácter ‘\n’ (que indica o fim de uma trama), a aplicação analisa os 6 primeiros caracteres da *string* armazenada, ou seja, verifica se o seu cabeçalho é igual a ‘\$GPGGA’. Se for igual, realiza um *parsing* à *string* e os dados (latitude, longitude, altitude e número de satélites) são passados para a *thread* A, que por sua vez os imprime na janela da aplicação. Se não for igual, estamos na presença de um dos outros três tipos de tramas do protocolo NMEA 0183 reconhecidas pelo recetor utilizado; nesse caso, a trama é descartada, ou seja, o conteúdo da *string* não é usado.

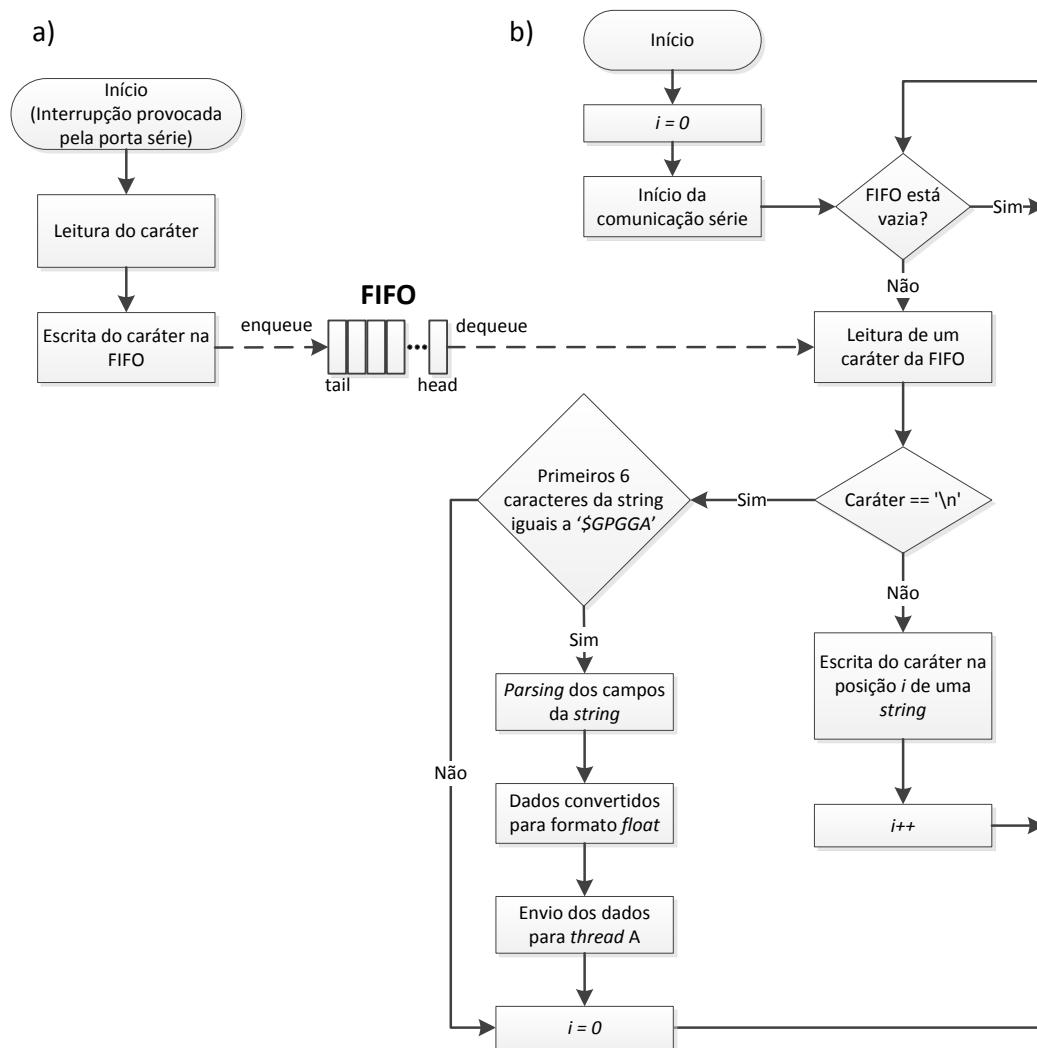
Foi implementada uma fila *First In, First Out* (FIFO) de caracteres entre a receção e a *thread* B, uma vez que se verificou que quando o *parsing* era realizado o seu tempo de execução era superior ao intervalo de tempo entre a receção de caracteres consecutivos (note-se que esta é desencadeada por interrupções ao sistema). A implementação da FIFO permite acomodar os caracteres que se perderiam aquando da execução do *parsing*. Sem ela, ocorreria perda de caracteres, tornando as tramas ilegíveis.

Durante o desenvolvimento da aplicação observou-se que os dados devolvidos pelo recetor GPS apresentavam pouca precisão (grande variação dos resultados, com discrepâncias entre valores devolvidos para mesma posição na ordem dos metros) mas alguma exatidão (valor médio desses resultados próximo dos reais valores numéricos das posições). Estas observações sugeriram a aplicação de filtragem passa-baixo através de média deslizante (*moving average* [33]). Nesta aplicação de teste, a filtragem foi realizada na *thread* A, através de um algoritmo que realiza a média das latitudes e das longitudes devolvidas pela *thread* B. Este algoritmo utiliza *arrays* de dimensão GPS\_READINGS (parâmetro configurável no ficheiro *header* da aplicação) para guardar os últimos GPS\_READINGS valores de latitude e longitude e apresenta os seus valores médios na janela. Uma avaliação preliminar conduziu a um valor de GPS\_READINGS igual a 3, que pareceu melhorar a precisão sem penalizar em demasia a latência na atualização dos resultados.

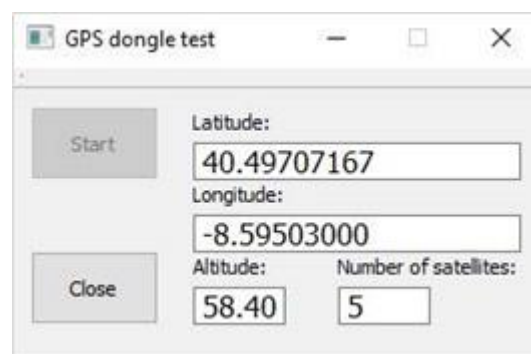
A Figura 3-7 mostra o aspeto da *interface* gráfica da aplicação, onde são exibidos os dados recebidos. Estão disponíveis dois botões: ‘Start’, para iniciar a execução da *thread* B, e ‘Close’, que destrói as *threads* e encerra a aplicação.

O código desta aplicação, designado *GPSdongleTest.pro*, encontra-se no CD que acompanha esta dissertação (vide Anexo C).





**Figura 3-6:** a) Interrupção da porta série; b) *Thread* de receção de dados de GPS (*Thread B*)



**Figura 3-7:** Interface da aplicação de aquisição de dados do recetor GPS

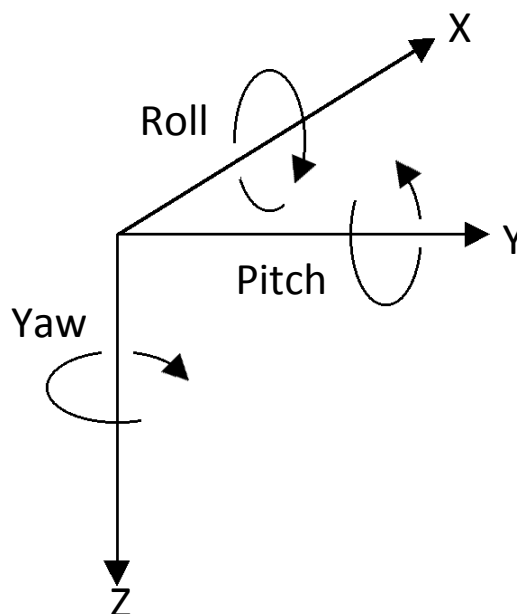
## 3.2 Detecção de orientação

### 3.2.1 Introdução

Os filtros HRTF traduzem a influência da interação das ondas sonoras com a cabeça do ouvinte na percepção da proveniência do som. Na aquisição das HRTF, brevemente apresentada no capítulo 2.1 (Figura 2-6), a cabeça é colocada no centro de um referencial cartesiano, perfeitamente alinhada com os seus eixos.

O sistema de auralização escolhe o par de filtros HRTF em função dos ângulos de azimuth e elevação da fonte virtual relativamente à cabeça do ouvinte. Esses ângulos dependem não só da posição do ouvinte (detetada pelo sistema de GPS descrito na secção 3.1) mas também da orientação do referencial associado à cabeça. Esta altera-se ao longo do tempo, de forma deliberada ou inadvertida. Em qualquer caso, e por mais ligeira que seja a alteração, deve ser levada em conta para uma correta percepção da proveniência do som. Por isso, é necessário detetar os movimentos do referencial da cabeça (vide Figura 3-8):

- ***Yaw*** - Rotação em relação ao eixo dos  $zz$ ;
- ***Pitch*** - Rotação em relação ao eixo dos  $yy$ ;
- ***Roll*** - Rotação em relação ao eixo dos  $xx$ ;



**Figura 3-8:** Rotações em relação aos eixos coordenados

Dados estes ângulos de rotação, a biblioteca de auralização AcousticAVE, apresentada na secção 2.1.6, é capaz de traduzi-los (tendo em conta também as posições da fonte e do ouvinte) em termos de azimuth e elevação, escolher as correspondentes HRTF [14] e aplicá-las em tempo real, evitando assim que a fonte virtual pareça acompanhar a rotação da cabeça do ouvinte.

### 3.2.2 Sensor inercial

Para detetar a orientação da cabeça do ouvinte, usou-se o sensor inercial (IMU) mostrado na Figura 3-9 [34]. Este dispositivo é capaz de medir a orientação em torno dos seus três eixos (*pitch*, *yaw* e *roll*) e apresenta grande precisão e baixa latência, o que o torna ideal para as necessidades do projeto.



**Figura 3-9:** *InertiaCube3* da Intersense (fonte: <https://est-kl.com/it/manufacturer/intersense/wireless-inertiacube-3.html>)

As suas pequenas dimensões permitem integrá-lo facilmente na faixa dos auscultadores. As principais características do sensor são apresentadas na Tabela 3-5 [35].

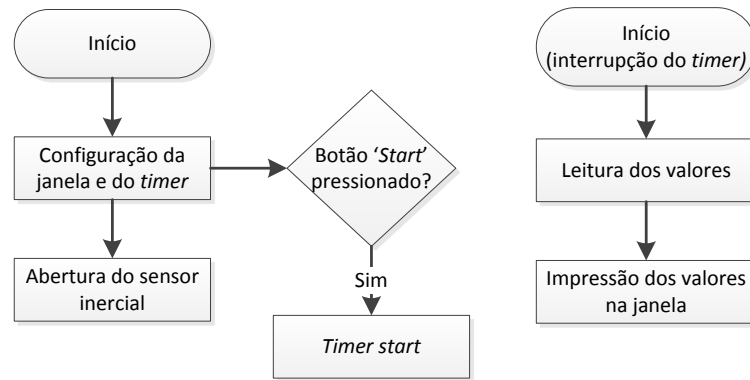
**Tabela 3-5:** Características do sensor *InertiaCube3*

Interface	USB
Latência	4 ms (via USB)
Frequência máxima de refrescamento	180Hz
Graus de liberdade	3 eixos ( <i>Yaw</i> , <i>Pitch</i> , and <i>Roll</i> )
Alcance angular	360° (Todos os eixos)
Precisão	<i>Yaw</i> : 1°; <i>Pitch</i> e <i>Roll</i> : 0.25° (à temperatura de 25°)
Velocidade angular máxima	1200° por segundo

### 3.2.3 Aplicação para aquisição de dados do sensor inercial

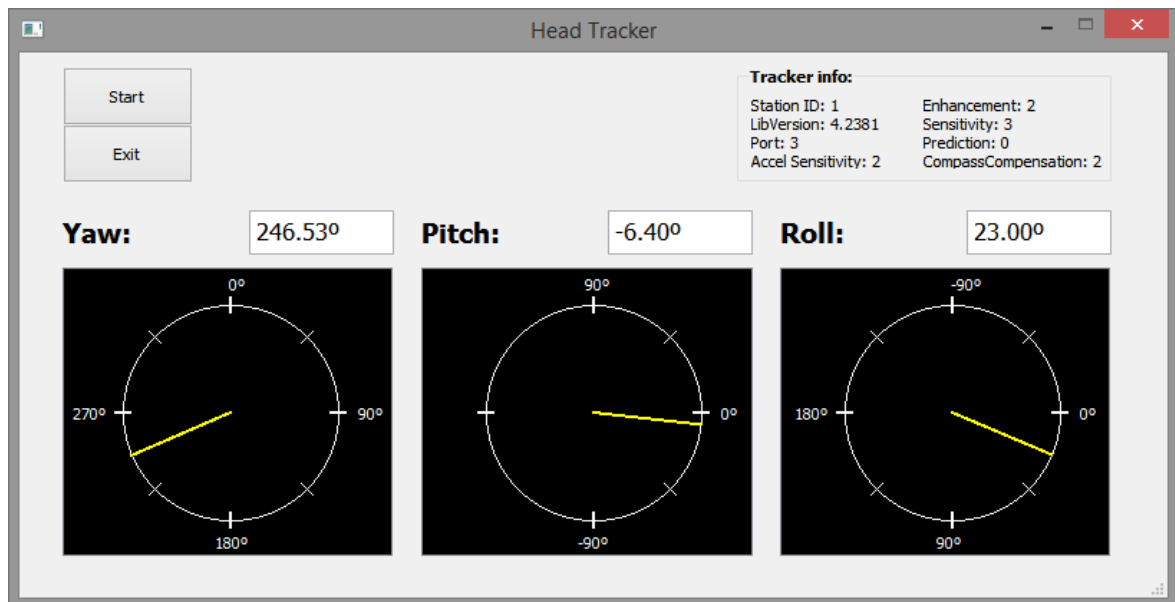
Este dispositivo apresenta uma API que permite operá-lo, possibilitando a sua configuração e aquisição dos valores devolvidos [36]. A API guarda os valores devolvidos pelo sensor em estruturas de dados.

Desenvolveu-se uma aplicação em linguagem C++ que mostra os valores de *yaw*, *pitch* e *roll* numa janela – vide Figura 3-11. Esta aplicação utiliza a configuração padrão do dispositivo e tem um funcionamento bastante simples, ilustrado no fluxograma da Figura 3-8: um *timer* com *timeout* de 6ms (aproximadamente correspondente aos 180Hz de frequência máxima de refrescamento do dispositivo) causa sucessivas interrupções para leitura dos dados das estruturas e sua impressão na *interface* da aplicação.



**Figura 3-10:** Funcionamento da aplicação de aquisição de dados do sensor inercial

Como se vê na Figura 3-11, a interface da aplicação permite observar os dados pretendidos (*yaw*, *pitch* e *roll*); o botão ‘Exit’ permite parar o *timer* e encerrar a aplicação.



**Figura 3-11:** Aplicação de aquisição de dados do sensor InertiaCube3

Devido a ter sido usada a configuração padrão, a calibração do dispositivo (para compensação de campos magnéticos estáticos) é feita automaticamente com recurso à bússola presente nele [36].

O código desta aplicação, designado *InertiaCubeTest.pro*, encontra-se no CD que acompanha esta dissertação (vide Anexo C).



## 4. Integração dos sistemas

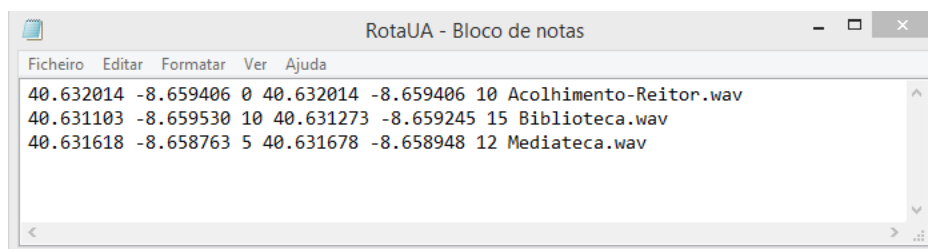
Aqui são apresentadas duas aplicações desenvolvidas em linguagem C++ para implementar os blocos ‘Gestor de rota’ e ‘Reprodução’ (vide Figura 1-10). Estas aplicações integram todos os outros elementos estudados anteriormente neste documento e, em conjunto, constituem a meta deste projeto.

### 4.1 Gestor de rota

Esta aplicação permite especificar uma coleção de fontes virtuais, que constituirão a rota de visita. Cada fonte virtual é caracterizada pelos seguintes conjunto de atributos:

- **Coordenadas geográficas horizontais (latitude e longitude, em graus) do ponto correspondente (‘S’).**
- **Altura (em metros) do ponto ‘S’ em relação ao ouvinte, considerado na região de *trigger*.**
- **Coordenadas geográficas horizontais (latitude e longitude, em graus) do centro da região *trigger* (‘TP’).**
- **Raio (em metros) da região de *trigger*.**
- **Nome do ficheiro de áudio monofónico associado.**

Toda esta informação é guardada num ficheiro em formato de texto (.txt), que designaremos ‘ficheiro de rota’. Cada linha do ficheiro corresponde a uma fonte virtual e compreende os atributos enumerados – vide exemplo na Figura 4-1.



**Figura 4-1:** Ficheiro de rota

A aplicação baseia-se numa *thread* que gere a interface gráfica, a criação e escrita dos dados no ficheiro de rota. Uma segunda *thread* (igual à *thread* de *parsing* de dados do recetor GPS descrita na secção 3.1.3) permite que a especificação da latitude e longitude (quer dos pontos ‘S’ quer dos pontos ‘TP’) seja feita

por medição ‘*in situ*’, desde que o recetor GPS esteja disponível. Implementa assim o bloco opcional indicado no diagrama da Figura 1-10.

O aspeto da *interface* da aplicação é apresentado na Figura 4-2: a parte superior (assinalada pelo número 1) controla o lançamento da *thread* que faz a gestão do recetor GPS (botão ‘*Connect GPS*’) e apresenta os dados recebidos. É na parte intermédia (assinalada pelo número 2) que os atributos de cada fonte sonora virtual são inseridos. Podem sê-lo manualmente ou, no caso das coordenadas de ‘*S*’ ou ‘*TP*’, automaticamente, por captura dos dados recebidos por GPS (botão ‘*Capture coordinates*’); nas *checkboxes* adjacentes deve ser antecipadamente selecionada a opção pretendida (‘*S*’ ou ‘*TP*’). Após a inserção de todos os dados de uma fonte sonora virtual, é possível, usando os comandos da parte inferior (assinalada pelo número 3), acrescentar esses dados ao ficheiro de rota (botão ‘*Add position*’), guardar o ficheiro de rota (botão ‘*Save route file*’) ou terminar a aplicação (‘*Close*’). O número de fontes inseridas no ficheiro de rota (‘*Number of sources added*’) é atualizado continuamente.

À semelhança da aplicação desenvolvida em 3.1.3, aqui também se pretendeu por melhorar a precisão dos valores devolvidos pelo recetor GPS. Então sempre que for usado o recetor GPS, para aquisição de ‘*S*’ e/ou ‘*TP*’, eles são filtrados de igual modo (*moving average*). Contudo, o número de amostras usadas para a filtragem é selecionável diretamente na interface da aplicação (*spinbox* denominada ‘*Window length*’ da parte 1 da interface), podendo-se, assim, definir uma aquisição de filtragem mais longa, permitindo ao utilizador permanecer na mesma posição melhorando a precisão dos valores das posições obtidos, uma vez que esta aplicação não exige compromissos com latência.

O código desta aplicação, designado *RouteManager.pro*, encontra-se no CD que acompanha esta dissertação (vide Anexo C).

**Route manager**

**Current position (GPS):**

Latitude: 40.4971000 Longitude: -8.5950783 Altitude (m): 66.70 Satellites used: 8 Window length: 5

Connect GPS

**Moving average filter**

Capture coordinates ☐ To Source position ☒ To Trigger Point position

**Source**

Latitude: 40.4969350 Longitude: -8.5947383 Height (m): 5 (Relative to the listener)

**Trigger point**

Latitude: 40.4970967 Longitude: -8.5951033 Trigger radius (m): 10

**Audio file name:**

audio.wav

Number of sources added: 1

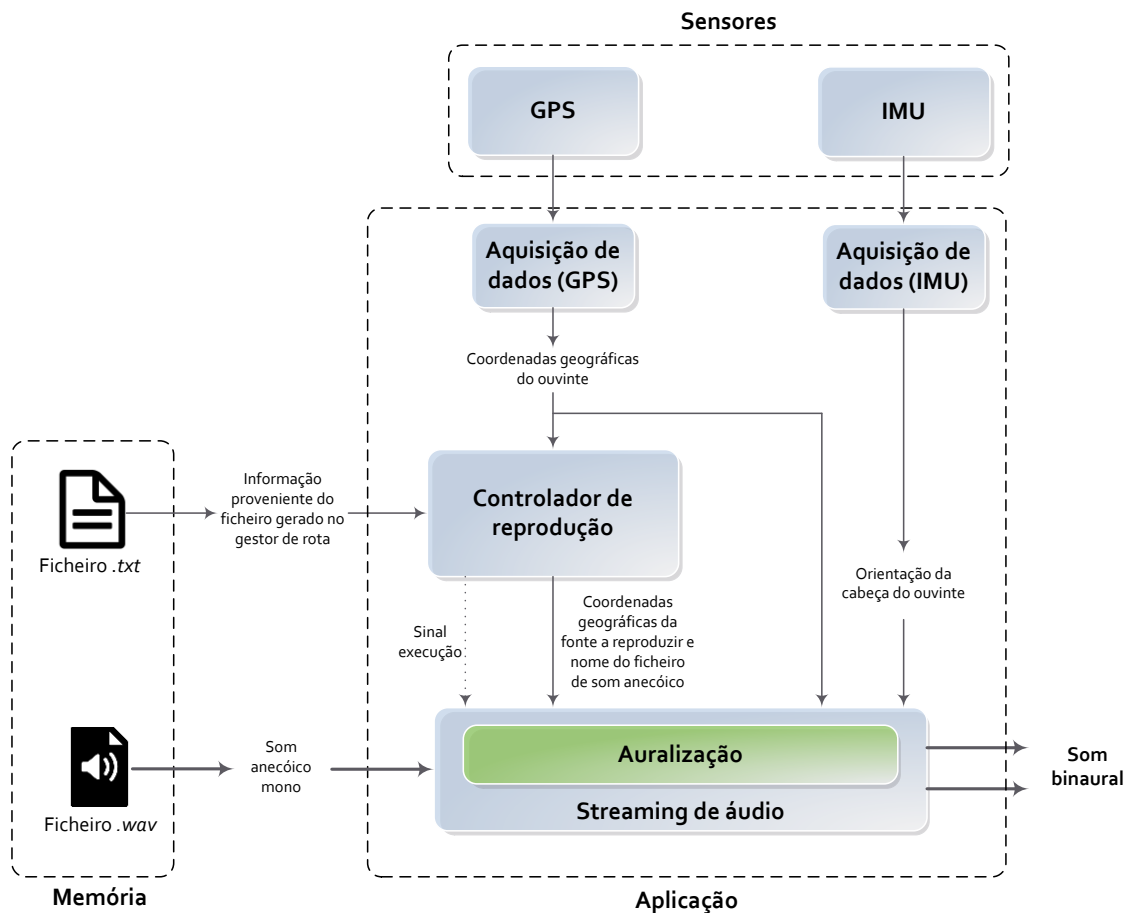
Add position Save route file Close

**Figura 4-2:** Interface gráfica da aplicação de gestão de rota



## 4.2 Controlo de reprodução

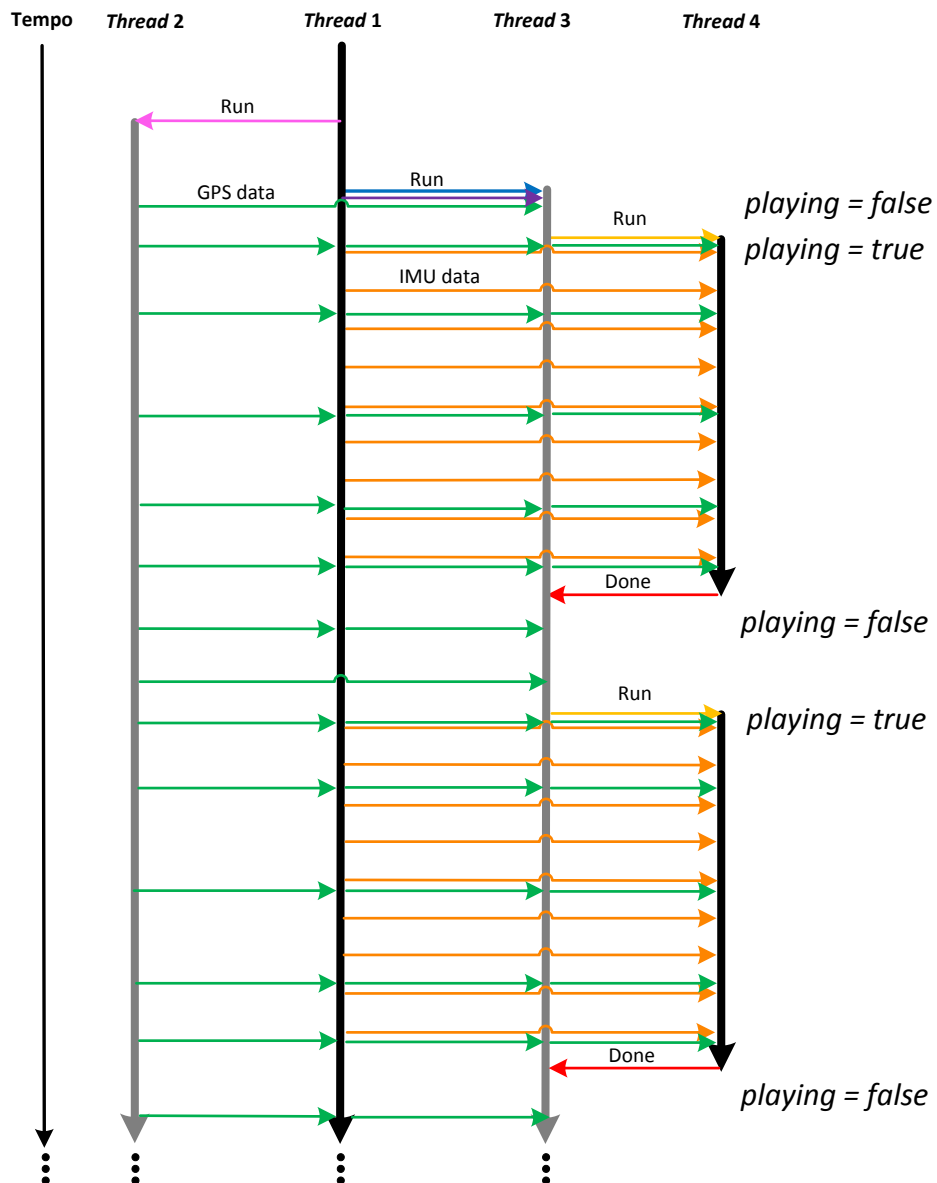
A Figura 4-3 apresenta novamente todos os elementos envolvidos na aplicação e as trocas de dados entre eles, dando agora especial ênfase ao sistema de reprodução. A articulação entre o bloco ‘*Streaming* de áudio’, que se baseia no motor de auralização, e os restantes elementos do sistema é garantida pelo bloco funcional aqui designado ‘Controlador de reprodução’. Este é responsável por interpretar o conteúdo do ficheiro de rota e, combinando esta informação com os dados provenientes do recetor GPS, definir em que momentos e sob que condições deve ser ativado o *streaming* de áudio. A área denominada por ‘Aplicação’, na Figura 4-3, representa o bloco ‘Reprodução’ – vide Figura 1-10.



**Figura 4-3:** Elementos integrados na aplicação de reprodução e suas trocas de informações

### 4.2.1 Implementação e funcionamento

A aplicação recorre a 4 *threads* [31], uma vez que os processos que a integram apresentam eventos em *loop* e necessitam de transmitir informação entre eles. A Figura 4-4 ilustra a execução dessas 4 *threads* ao longo do tempo e a troca de informação que ocorre entre elas. Para facilitar a ilustração, o eixo temporal não obedece a uma escala linear. Por exemplo, atendendo às frequências de refrescamento especificadas pelos fabricantes dos recetores, por cada vez que são recebidos dados do recetor GPS ('GPS data') poderíamos ter 180 chegadas de dados do sensor inercial ('IMU data').



**Figura 4-4:** Diagrama temporal de execução das *threads* e principais trocas de informação entre elas

Segue-se um resumo, apoiado na Figura 4-4, das funções de cada *thread*.

**Thread 1:**

- Gestão da interface gráfica da aplicação.
- Verificação da presença do sensor inercial e recetor GPS no sistema (ligação física por USB ao computador).
- Receção dos dados provenientes do sensor inercial e sua passagem à *thread* 4 (setas cor-de-laranja - ‘IMU data’); a aplicação desenvolvida para este efeito encontra-se descrita na secção 3.2.3.
- Leitura do ficheiro de rota e passagem dos dados à *thread* 3 (seta púrpura).
- Lançamento das *threads* 2 e 3 (setas cor-de-rosa e azul).

**Thread 2** (corresponde à *thread B* descrita em secção 3.1.3):

- *Parsing* das tramas \$GPGGA recebidas e passagem de dados às *threads* 1, 3 e 4 (setas verdes – ‘GPS data’).

**Thread 3** (constitui o ‘Controlador de reprodução’):

- Lançamento da *thread* 4 (seta amarela), condicionado pela localização do ouvinte.

**Thread 4:**

- Execução da auralização e *streaming* de áudio (com base na aplicação descrita na secção 2.2.2).

Na *thread* 1, após confirmação da presença dos sensores e abertura do ficheiro de rota, pode dar-se início à execução da *thread* 3 premindo o botão ‘Start’ na interface principal da aplicação – vide Figura 4-11.

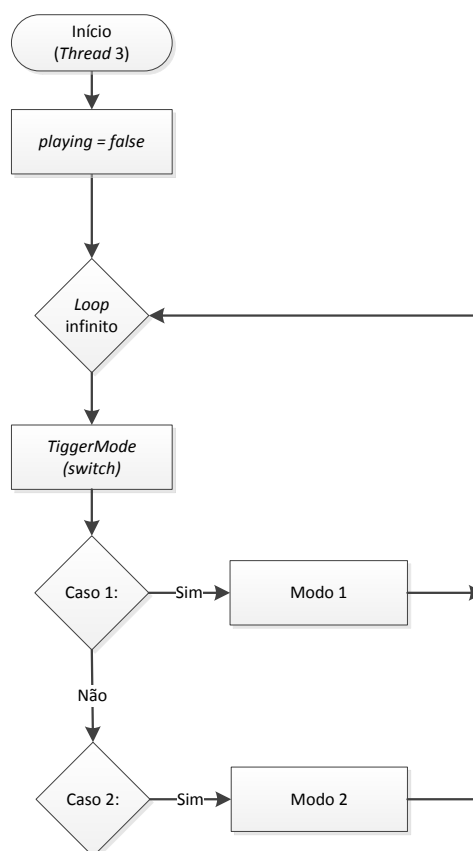
Como a *thread* 3 é responsável pelo lançamento da *thread* 4, começa por receber da *thread* 1 todas as fontes de áudio virtuais com todos os dados associados. Estes dados são armazenados sob a forma de um *array* de estruturas, representando cada estrutura uma fonte de som virtual com os seus atributos. Para este efeito, definiu-se um tipo de estrutura designado *Source\_Info*, com os seguintes campos:

- **Coordinates** - Coordenadas geográficas da fonte sonora virtual (‘S’).
- **CoordinatesTP** - Coordenadas geográficas do ponto central da região de *trigger* (‘TP’).
- **TriggerRadius** - Raio que define a região de *trigger* (círculo centrado no respetivo *TP*).
- **AudioFileName** - Nome do ficheiro de áudio monofónico da fonte sonora virtual.
- **Reproduced** – *Flag* que indica se esta fonte sonora virtual sonora virtual já foi reproduzida.
- **OutTriggerRadius** – *Flag* que indica se o ouvinte saiu da região de *trigger*.

As variáveis *Coordinates* e *CoordinatesTP* são declaradas como objetos da classe *QGeoCoordinates* disponibilizada no Qt Creator, que permite manipular coordenadas geográficas de forma simples e prática. A variável *TriggerRadius* é declarada como sendo do tipo *float* e representa distância em metros; a variável *AudioFileName* é uma *string*; os campos *Reproduced* e *OutTriggerRadius* são do tipo *boolean*, sendo *Reproduced* inicializada com o valor *false* (fonte ainda não reproduzida) e *OutTriggerRadius* com o valor *true*

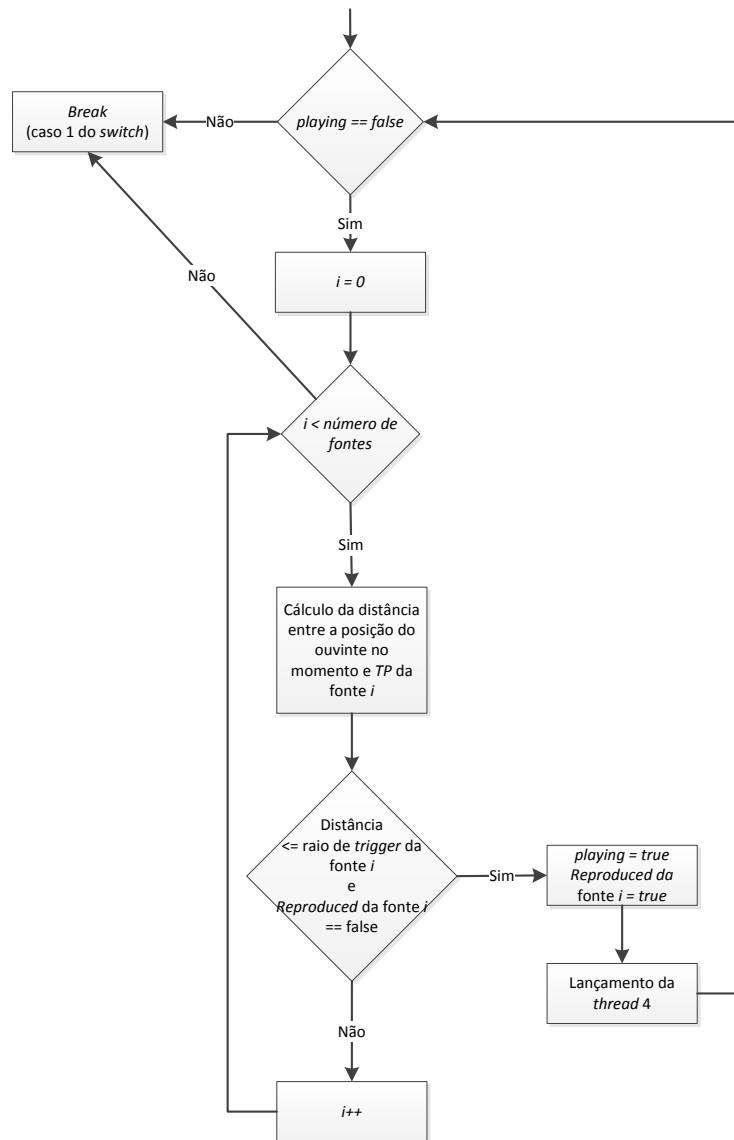
(ouvinte fora da região de *trigger*), para cada fonte sonora virtual. Os primeiros quatro campos da estrutura provêm dos dados contidos no ficheiro de rota.

A *thread* 3 recebe continuamente as coordenadas do ouvinte, e em função delas determina se deve ser iniciada a reprodução dos conteúdos correspondentes. Foram definidos dois modos de *trigger* selecionáveis através do menu de configuração da aplicação. O modo escolhido é registado na variável *TriggerMode*, como ilustra a Figura 4-5. Previamente também é configurada uma variável de estado booleana, *playing*, usada para definir se é permitido o lançamento da *thread* 4, impedindo-o sempre que tiver o valor *true*. Inicialmente toma o valor *false* e fica *true* sempre que está em curso a reprodução de uma fonte (Figura 4-4). Assim, não permite a reprodução de várias fontes em simultâneo nem o lançamento repetido da *thread* 4 para a mesma fonte.



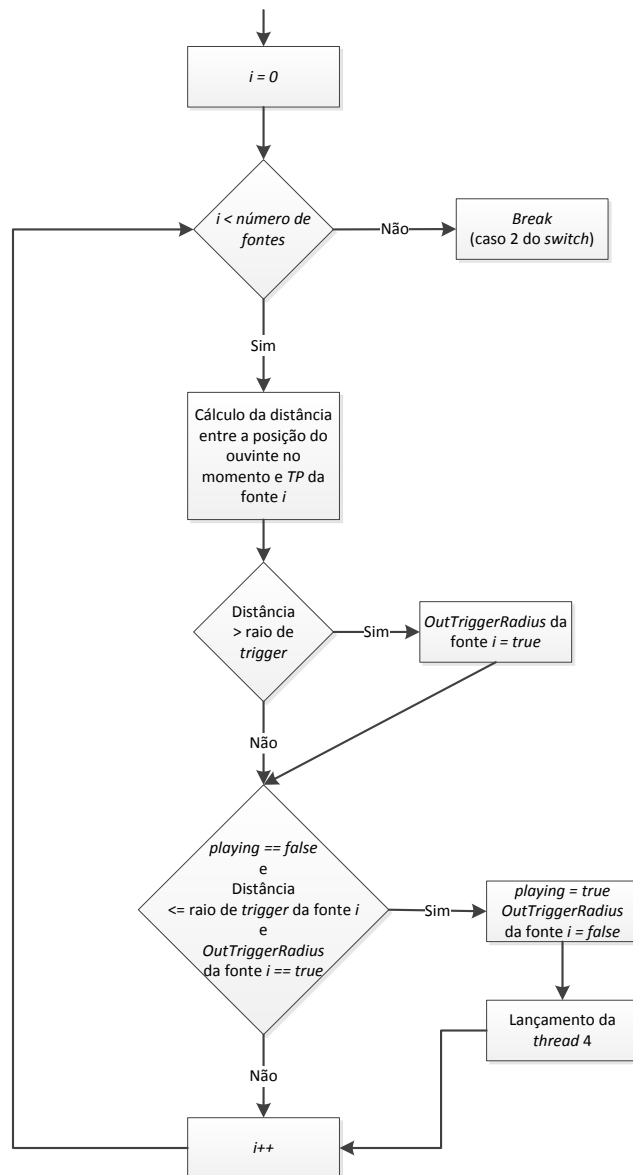
**Figura 4-5:** Seleção do modo de *trigger*

No primeiro modo de *trigger*, o bloco ‘Auralização’ é executado apenas uma vez por cada fonte virtual (quando o ouvinte entra na sua região de *trigger*), não sendo permitida reprodução repetida. O *array* de fontes é percorrido e são atualizadas as distâncias ao ponto de *trigger* de cada fonte. Como mostra a Figura 4-6, quando o ouvinte se encontra uma distância inferior ao raio de *trigger* respectivo, a reprodução dessa fonte é permitida desde que não tenha já ocorrido e não haja nenhuma outra fonte a ser reproduzida nesse momento. O cálculo dessa distância (bem como o de todas as distâncias neste projeto) é feito com recurso a um método presente na classe *QGeoCoordinates* [37] do Qt Creator, que toma a distância entre dois pontos de uma superfície esférica como sendo o comprimento do arco de circunferência mais curto que os une nessa superfície (*great-circle distance* [38]). É de notar que este cálculo usa um modelo da Terra completamente esférico, com raio constante, e por isso os valores de altitude não são tidos em conta; somente são usados para efeitos de auralização.



**Figura 4-6:** Modo 1 de *trigger*

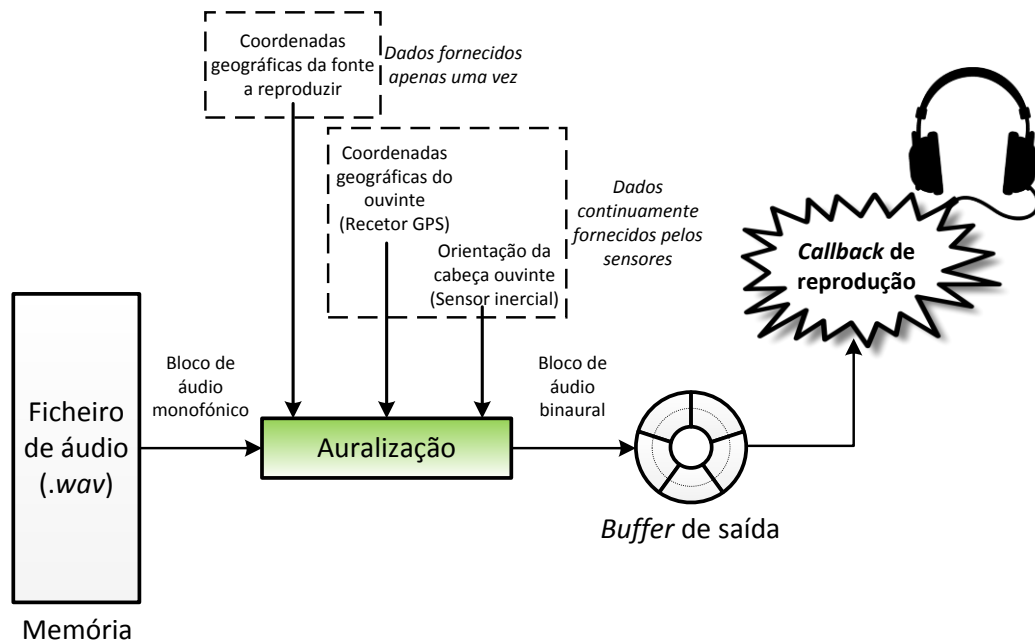
No segundo modo de *trigger*, o bloco ‘Auralização’ é executado sempre que o ouvinte entra na região de *trigger* de alguma fonte virtual, permitindo a repetição da reprodução dos mesmos conteúdos áudio. Neste modo, também é percorrido o *array* de fontes, sendo atualizadas as distâncias ao ponto de *trigger* de cada uma. Como mostra a Figura 4-7, quando o ouvinte se encontra uma distância inferior ao raio de *trigger* respectivo, a reprodução dessa fonte é permitida desde que não tenha acabado de ser reproduzida (i.e. não é permitido *looping*) e não haja nenhuma outra fonte a ser reproduzida nesse momento. As *flags OutTriggerRadius* são usadas para assinalar entrada (ou re-entrada) e evitar *looping*.



**Figura 4-7:** Modo 2 de *trigger*

Em ambos os modos, a *flag playing* é colocada a *false* quando a *thread* 4 for finalizada (sinal ‘Done’ – vide Figura 4-4), permitindo assim um novo *trigger* se as condições do modo que esteja a ser usado forem satisfeitas. A posição do ouvinte é continuamente atualizada com dados provenientes da *thread* 2 (sinal ‘GPSdata’ – vide Figura 4-4).

A *thread* 4 (Figura 4-8) corresponde essencialmente ao sistema estudado na secção 2.2.2, com a diferença de a posição e a orientação da cabeça do ouvinte serem fornecidas (e continuamente atualizadas) pelas *threads* que executam a gestão dos sensores (mensagens ‘IMUdata’, da *thread* 1, e ‘GPSdata’, da *thread* 2) – vide Figura 4-4 – e os dados da fonte sonora virtual (posição e nome do ficheiro áudio) serem fornecidos (à cabeça, pois nunca sofrem alteração) pela *thread* 3 (que é executada uma vez para cada fonte).



De modo a aumentar a funcionalidade da aplicação, passou-se a utilizar ficheiros de áudio no formato .wav, que, por ser o mais difundido, é facilmente acessível a qualquer utilizador, mesmo sem conhecimentos técnicos na área. Os ficheiros áudio devem ser armazenados na mesma pasta do ficheiro de rota. A alocação de memória foi gerida com o auxílio de uma biblioteca externa, a Libsndfile [39] (Anexo B). Trata-se de uma biblioteca de acesso livre, escrita em C, que permite abrir uma grande variedade de formatos de ficheiro de áudio. Esta biblioteca é então responsável pela leitura e verificação do formato do ficheiro de áudio monofónico e extração dos seus parâmetros (configuração descrita no Anexo D). No caso de alguma verificação falhar, o erro é comunicado à *thread* 1, que o apresenta ao utilizador.

Após um estudo relativamente superficial da biblioteca AcousticAVE, procurou-se eliminar os ficheiros e segmentos de código desnecessários tendo em conta que em ambiente exterior são dispensáveis as funcionalidades relativas à modelação do ambiente acústico (reflexões, reverberação...).

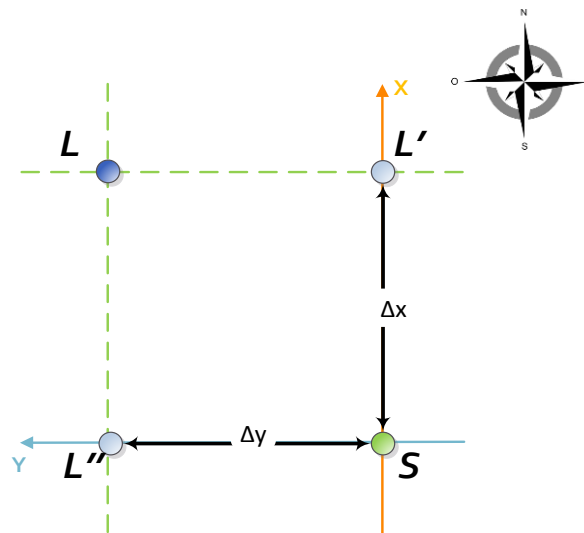
O algoritmo de auralização contempla um parâmetro de atenuação da onda sonora devido à distância, proporcional ao quadrado da distância entre a posição do ouvinte e da fonte sonora virtual. Nesta aplicação,

porém, tendo em vista otimizar o conforto auditivo e reduzir os constrangimentos no traçado de rotas, considerou-se vantajoso desativar a variação de intensidade com a distância; a experiência imersiva aproveita apenas a noção de direção. Aplicou-se assim um fator de atenuação unitário.

De resto, o AcousticAVE manteve as suas funções nucleares e as 4 bases de dados de HRTF, preservando assim as suas características essenciais para o projeto: motor de auralização com funcionamento em tempo real.

Para referenciar as posições do ouvinte e da fonte sonora virtual, a AcousticAVE utiliza um sistema cartesiano em metros. Os eixos usam a orientação comum dos programas de arquitetura *Computer Aided Design* (CAD):  $x$  = Norte,  $y$  = Oeste,  $z$  = altura [14].

Considere-se uma fonte sonora virtual  $S$  com coordenadas horizontais ( $lat_S$  e  $lon_S$ ) e um ouvinte  $L$  com coordenadas horizontais  $L(lat_L, lon_L)$ . Considere-se ainda que a fonte sonora virtual  $S$  se encontra sempre na origem do referencial  $xy$  ( $z$  será um parâmetro previamente fornecido pelo ficheiro de rota, já que é a altura da fonte). As projeções da posição do ouvinte  $L$  nos eixos  $x$  e  $y$  terão coordenadas  $L'(lat_L, lon_S)$  e  $L''(lat_S, lon_L)$ , respetivamente, como mostra a Figura 4-9.

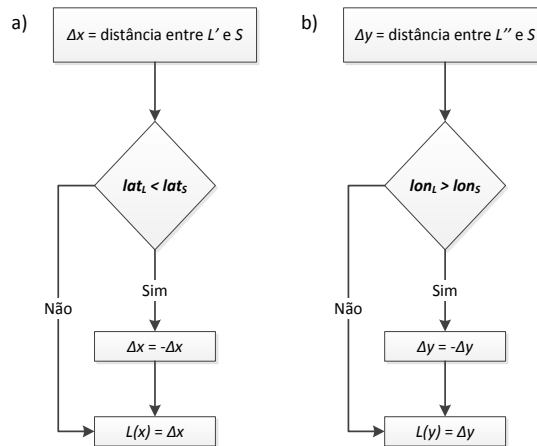


**Figura 4-9:** Posições relativas do ouvinte e da fonte sonora

Para obter a posição  $L(x,y)$  do ouvinte num referencial cartesiano, foi incluído na *thread* 4 um algoritmo que converte as coordenadas geográficas da posição do ouvinte fornecidas pelo recetor GPS. Este algoritmo calcula a distância em metros entre  $L'$  e  $S$  e entre  $L''$  e  $S$ , mais uma vez com recurso à classe QGeocoordinates do Qt Creator. As distâncias obtidas são em valor absoluto; para determinar os valores relativos (com sinal), o algoritmo efetua as comparações descritas na Figura 4-10.

Como o eixo  $yy$  apresenta um sentido oposto ao sentido positivo da longitude, a condição de comparação das longitudes entre o ouvinte e a fonte sonora virtual é inversa à de comparação das latitudes.





**Figura 4-10:** a) Cálculo de  $L(x)$  b) Cálculo de  $L(y)$

Para os ângulos de orientação da cabeça, medidos em radianos, a biblioteca AcousticAVE usa um sistema de coordenadas comum em sensores inerciais, onde (sendo  $x$  = Norte,  $y$  = Este e  $z$  = baixo – Figura 3-8) [14]:

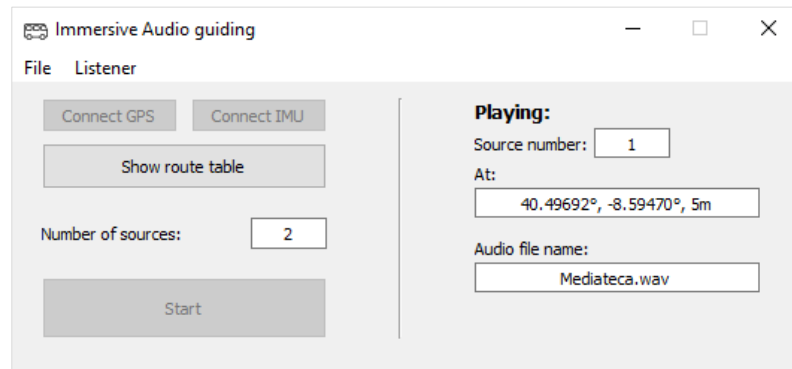
- **roll** > 0 inclinação para a direita, **roll** < 0 inclinação para a esquerda;
- **pitch** > 0 inclinação para cima, **pitch** < 0 inclinação para baixo;
- **yaw** > 0 rotação para a direita, **yaw** < 0 rotação para a esquerda, com **yaw** = 0 para a direção Norte;

O sensor usado neste projeto utiliza o mesmo sistema de coordenadas, podendo assim os valores da orientação da cabeça do ouvinte ser passados diretamente da *thread 1* para a *thread 4*.

O código desta aplicação, designado *ImmersiveAudioGuiding.pro*, encontra-se no CD que acompanha esta dissertação (vide Anexo C).

### 4.2.2 Interface da aplicação

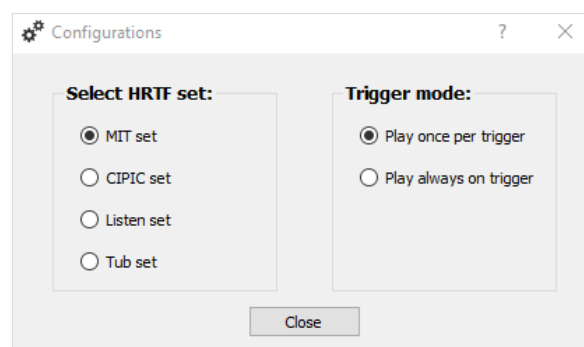
Procurou-se criar uma interface simples e prática, com acesso a todos os dados provenientes do ficheiro de rota e dos sensores (recetor GPS e sensor inercial), bem como ao estado e distância de cada fonte de som virtual. A interface principal da aplicação, apresentada na Figura 4-11, permite também conectar os dispositivos.



**Figura 4-11:** Interface principal da aplicação da *Immersive Audio guiding*

Ao pressionar os botões ‘*Connect GPS*’ e ‘*Connect IMU*’ dá-se início ao funcionamento dos sensores e tratamento dos dados deles provenientes, desde que estejam ligados fisicamente por USB ao computador; se tal não acontecer, serão mostradas mensagens de erro. Consequentemente, no caso de o recetor GPS se encontrar presente e se pressionar o botão ‘*Connect GPS*’ dá-se início à execução da *thread 2* (Figura 4-4).

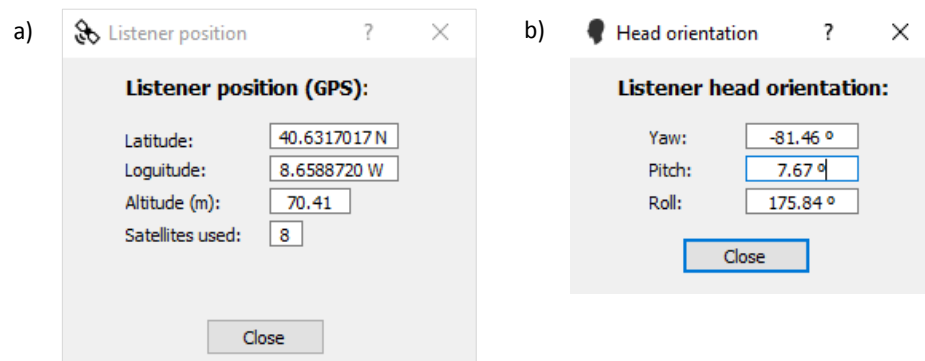
Através do separador ‘*File*’, podemos carregar o ficheiro da rota (‘*File→Load route file*’), alterar configurações da aplicação (‘*File→Configurations*’), e escolher o conjunto de HRTF usado para a auralização (no lado esquerdo da janela) ou o modo de *trigger* da aplicação (no lado direito da janela), como se mostra na Figura 4-12.



**Figura 4-12:** Janela do painel de configuração

Depois de aberto o ficheiro de rota, podemos ver na interface principal da aplicação o número de fontes virtuais carregadas para a aplicação (‘*Number of sources*’).

No separador ‘*Listener*’, temos duas opções que permitem ver os dados recebidos por cada sensor em tempo real, obtendo janelas com os seguintes aspetos:



**Figura 4-13:** a) Janela da posição do ouvinte b) Janela da orientação da cabeça do ouvinte

Após o ficheiro de rota ser carregado e os sensores estarem em funcionamento, o botão ‘*Start*’ na interface principal da aplicação aparecerá como ativo, permitindo ao utilizador dar início ao bloco ‘Controlador de reprodução’, ou seja, à execução da *thread 3*.

Usando o botão ‘*Show route table*’ da interface principal da aplicação, podemos ver todos os dados vindos do ficheiro de rota, o estado de reprodução de cada fonte virtual (campo ‘*Reproduced*’) e a distância do ouvinte até à posição de cada fonte virtual em tempo real (campo ‘*Distance to S.*’), como mostra a Figura 4-14. No caso do campo ‘*Reproduced*’, podemos ver se a fonte já foi ou não reproduzida (mensagens ‘*Yes*’ ou ‘*No*’) através do controlo feito na *thread 3*; poderá ser mostrada uma mensagem de erro (‘*Error*’) no caso de a Libsndfile falhar alguma validação na abertura do ficheiro de áudio.

	File name	Reproduced	S. Latitude (°)	S. Longitude (°)	Altitude (m)	T. Latitude (°)	T. Longitude (°)	Trigger radius (m)	Distance to S. (m)
1	Biblioteca.wav	No	40.4970156	-8.5947494	0.0	40.4971356	-8.5948900	10.0	29.83
2	StereoBass.wav	Error	40.4971267	-8.5951133	5.0	40.4970394	-8.5952161	5.0	1.25
3	Mediateca.wav	Yes	40.4968217	-8.5952356	0.0	40.4969178	-8.5951072	12.0	15.22

**Figura 4-14:** Janela de estado das fontes

Para concluir, na zona de ‘*Playing*’ que se encontra no lado direito da interface principal da aplicação, no caso da *thread 4* ter sido lançada, são apresentados os dados mais relevantes da fonte sonora que se encontra a ser reproduzida; caso contrário, os campos aparecem em branco.



## 5. Testes e validação

Descreve-se aqui o conjunto de testes levados a cabo com o objetivo de avaliar o funcionamento da aplicação de reprodução (vide Figura 4-3) e validar o sistema no seu todo.

### 5.1 Conceção e preparação dos testes

Criou-se uma rota com 3 fontes sonoras virtuais dentro do *campus* da Universidade de Aveiro, como mostra a Figura 5-1. Preferiu-se uma rota pedestre a rota para veículos motorizados, uma vez que torna a logística de testes mais simples, sem aparentemente comprometer a qualidade dos mesmos. De facto, envolvem distâncias mais curtas e trajetos menos previsíveis, tornando os percursos pedestres muito mais exigentes em termos de exatidão e precisão na deteção da posição. Na Tabela 5-1 apresentam-se todos os parâmetros geográficos desta rota.



Figura 5-1: Traçado da rota de teste

**Tabela 5-1:** Configuração da rota de teste

	Coordenadas da fonte (S)			Coordenadas do ponto de <i>trigger</i> (TP)		Raio de <i>trigger</i> (m)
	Latitude (°)	Longitude (°)	Altura (m)	Latitude (°)	Longitude (°)	
<b>1</b>	40.632014	-8.659406	0	40.632014	-8.659406	10
<b>2</b>	40.631103	-8.659530	10	40.631273	-8.659245	15
<b>3</b>	40.631618	-8.658763	5	40.631678	-8.658948	12

A Tabela 5-2 indica os ficheiros áudio monofónicos associados às fontes. Trata-se de ficheiros em formato *.wav*, com frequência de amostragem de 44.1 kHz, 16bits por amostra e com uma duração suficientemente curta para o ouvinte, no seu percurso à pé, não entrar numa nova região de *trigger* sem ter acabado a reprodução do conteúdo anterior. Considerou-se que essa situação não seria conveniente, embora a aplicação não permita reproduções sobrepostas.

**Tabela 5-2:** Ficheiros áudio da rota de teste

	Nome do ficheiro de áudio	Duração (s)
<b>Fonte 1</b>	Acolhimento-Reitor.wav	38
<b>Fonte 2</b>	Biblioteca.wav	45
<b>Fonte 3</b>	Mediateca.wav	19

Refiram-se ainda as configurações (vide Anexo D) utilizadas para o tamanho do *buffer* de saída (*OBUF\_BLOCKS* = 5), amostras por bloco (*FRAMES\_PER\_BLOCK* = 256, o que corresponde aproximadamente aos 6ms de refrescamento do sensor inercial) e tamanho da janela de filtragem por *moving average* (*GPS\_READINGS* = 3).

É de notar que a fonte S1 foi deliberadamente configurada para coincidir com o centro da região de *trigger* ao contrário das fontes S2 e S3, situadas claramente fora da região de *trigger*.

No primeiro conjunto de testes, utilizou-se o modo de *trigger* 1 (sem repetição). Pediu-se aos sujeitos que avaliassem qualitativamente (3 níveis de resposta: bom, médio e mau) a experiência sob os seguintes parâmetros:

- **Trigger do áudio a uma distância aparentemente correta.**
- **Sensação do som provir da localização correta.**

- Para S1 (fonte coincidente com o centro da região de *trigger*), esta última avaliação foi feita em duas situações: longe (>5m) e perto (<5m) da fonte.

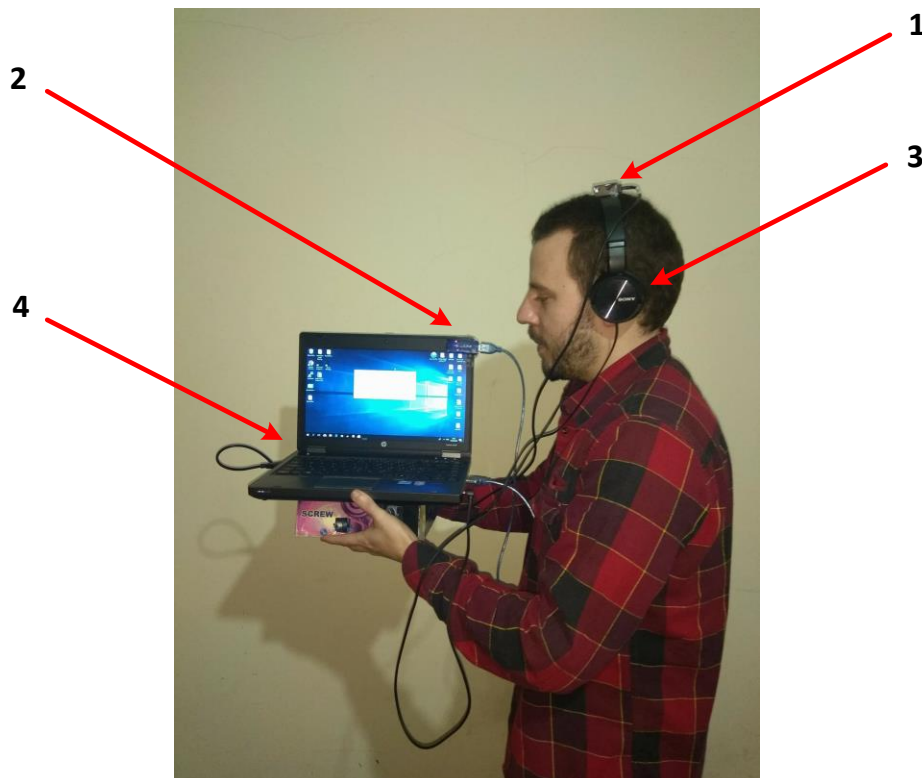
No segundo conjunto de testes, realizado apenas na região de *trigger* da fonte S2, utilizou-se o modo de *trigger* 2 (com repetição). Pediu-se aos sujeitos que verificassem se ocorria:

- **Repetição de reprodução do conteúdo áudio sem reentrada** (após o fim da primeira reprodução e ficando parados dentro da região de *trigger*) em duas situações:
  - Longe ( $> 5\text{m}$ ) do limite da região de *trigger*.
  - Perto ( $< 5\text{m}$ ) do limite da região de *trigger*.
- **Repetição de reprodução aquando da reentrada na região de *trigger*.**

Convidaram-se dez sujeitos escolhidos aleatoriamente (6 homens e 4 mulheres na faixa etária de 20 a 35 anos, sem problemas auditivos relatados) foram convidados percorrer a rota usando o sistema. Os sujeitos foram previamente informados dos dados de configuração associados a cada fonte.

A Figura 5-2 mostra o equipamento utilizado pelos sujeitos nos testes:

1. **Sensor inercial:** Foi usado o InertiaCube 3 para fornecer a orientação da cabeça do ouvinte à aplicação de reprodução.
2. **Recetor GPS:** Foi usado o recetor GPS Xucai GD75 para fornecer a posição do ouvinte à aplicação de reprodução.
3. **Headphones:** Foram usados *headphones* Sony MDR-ZX110, com uma gama de frequência de resposta de 12Hz aos 22kHz [40].
4. **Computador:** Aqui é onde é executada a aplicação de reprodução (bloco ‘Reprodução’).



**Figura 5-2:** Equipamento de teste

## 5.2 Resultados

Os dados qualitativos foram convertidos em valores numéricos conforme a Tabela 5-3.

**Tabela 5-3:** Chave de quantificação dos resultados

Apreciação qualitativa do ouvinte	Nível
Bom	3
Médio	2
Mau	1

No primeiro conjunto de testes (modo de *trigger* 1 – sem repetição de fontes), obtiveram-se os resultados apresentados nas Tabela 5-4 e Tabela 5-5:

**Tabela 5-4:** *Trigger* a uma distância aparentemente correta

Sujeito	1	2	3	4	5	6	7	8	9	10	Média	Desvio padrão	Moda
<b>Fonte 1</b>	3	2	3	3	2	2	3	2	2	2	2.4	0.52	2
<b>Fonte 2</b>	3	3	2	2	3	3	2	3	3	3	2.7	0.48	3
<b>Fonte 3</b>	3	3	2	3	3	2	2	3	2	3	2.6	0.52	3

**Tabela 5-5:** Sensação do som provir da localização correta

Sujeito		1	2	3	4	5	6	7	8	9	10	Média	Desvio padrão	Moda
Fonte 1	Longe da fonte (> 5m)	2	2	2	3	2	3	3	2	3	2.4	2.4	0.52	2
	Perto da fonte (< 5m)	1	1	2	1	1	2	1	2	1	1.4	1.4	0.52	1
Fonte 2		3	2	3	3	3	3	3	3	3	3	2.9	0.32	3
Fonte 3		3	3	2	3	3	2	2	3	3	3	2.7	0.48	3

Para testar o bom funcionamento deste modo de *trigger*, pediu-se aos ouvintes que reentrassem na região de *trigger* de S1 no fim da rota. Nenhum reportou repetição do conteúdo áudio, como seria de esperar.

No segundo conjunto de testes (modo de *trigger* 2 – com repetição de fontes), foi pedido aos ouvintes que, após ouvirem o conteúdo da fonte S2, esperassem dentro da região de *trigger* durante 60 segundos. Obtiveram-se os resultados da Tabela 5-6.

**Tabela 5-6:** Repetição de reprodução sem reentrada

Sujeito	1	2	3	4	5	6	7	8	9	10	Moda
<b>Fonte 2</b>	Longe do limite de <i>trigger</i> (> 5m)	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
	Perto do limite de <i>trigger</i> (< 5m)	Sim	Não	Sim	Não	Sim	Não	Não	Sim	Sim	Sim



Para testar o bom funcionamento deste modo de *trigger*, pediu-se aos ouvintes que saíssem e reentrassem na região de *trigger* de *S2*. Todos reportaram repetição do conteúdo áudio, como seria de esperar.

### 5.3 Análise crítica dos resultados

Com estes testes, comprovou-se que a aplicação se comporta genericamente como pretendido em ambos os modos de *trigger* definidos. A relativa imprecisão dos valores devolvidos pelo recetor GPS, acentuada pela reduzida escala da rota desenhada, é a principal fonte de imperfeições.

No primeiro modo de *trigger*, a reprodução iniciou-se sensivelmente às distâncias pretendidas; os resultados pareceram melhorar à medida que as regiões de *trigger* aumentavam, o que terá provavelmente a ver com a diminuição da relevância relativa do erro provocado pelo recetor GPS.

No que toca à sensação de proveniência do som, os resultados apenas se revelaram insatisfatórios no caso de fonte *S1* para distâncias inferiores a 5 metros. Isto é facilmente explicável: devido ao erro do recetor GPS, a posição do ouvinte medida pelo sistema oscila em redor da posição real; se a distância à fonte for da ordem de grandeza do erro (o que só ocorria nesta situação), o efeito auditivo torna-se inconsistente; no limite, o ouvinte pode mesmo passar virtualmente (como se verificou por vezes no teste) para o lado oposto da fonte. Nos restantes casos, os resultados são satisfatórios, e tanto melhores quanto mais longínquas as posições das fontes sonoras virtuais. De novo, este efeito pode ser justificado pela diminuição da relevância relativa do erro do recetor GPS.

No segundo modo de *trigger*, os resultados anómalos verificados perto do limite da região de *trigger* (áudio reproduzido novamente, como se tivesse ocorrido reentrada) podem explicar-se pela mesma razão: a oscilação da posição medida pelo recetor GPS gera reentradas virtuais do ouvinte. A distâncias do limite de *trigger* superiores a 5 metros, este erro deixa de ser observado.

O ficheiro de rota foi criado dias antes do teste recorrendo ao recetor GPS para estabelecer as coordenadas dos elementos da rota, o que acarreta algum viés, já que a constelação de satélites usada no cálculo das posições por parte do recetor GPS não é constante ao longo do tempo. No entanto, esse viés poderá não ser significativo e além disso, pode ser evitado por recurso a dados geodésicos de maior precisão.

Para além da influência do erro do recetor GPS, o instante de *trigger* pode ser afetado por latências ligadas não só ao próprio recetor de GPS (aquisição e filtragem de dados) como também à execução do algoritmo de *trigger*, ao processamento de auralização e ao *buffering* de áudio.

O tempo de atualização de dados do recetor GPS (da ordem dos  $10^3$ ms) predomina face ao tempo de processamento de auralização<sup>2</sup> (da ordem dos  $10^0$ ms) e ao tempo de *buffering* (atraso máximo do *buffer* de saída<sup>3</sup> aproximadamente 30ms). Podemos considerar assim um atraso um pouco maior que 1 segundo para o início da reprodução do áudio (pior caso). Este atraso de resposta também pode afetar, indiretamente, a sensação de proveniência do som. O efeito será tanto mais prejudicial quanto maior for a velocidade de locomoção do ouvinte e menor a escala da rota, mas normalmente estes fatores são interdependentes, com maiores velocidades naturalmente associadas a mais amplas escalas.

---

<sup>2</sup> Tempo adquirido no computador usado nos testes com recurso à biblioteca *time.h*.

<sup>3</sup>  $(5 * 256) / 44.1\text{kHz}$  com  $(OBUF\_BLOCKS * FRAMES\_PER\_BLOCK) / SAMPLE\_RATE$

No que toca ao sensor inercial, que tem efeito direto na sensação de proveniência do som, a situação é inversa: é o tempo de *buffering* que predomina, já que a latência do sensor inercial é de 4ms e a sua frequência máxima de atualização é 180Hz.



## 6. Conclusões e trabalho futuro

Pode considerar-se que os objetivos essenciais do trabalho foram atingidos, na medida em que se desenvolveu um protótipo funcional capaz de reproduzir conteúdos áudio pré-gravados criando o efeito de espacialização pretendido em tempo real e em ambiente *outdoor*.

Os testes preliminares efetuados indicaram que a imprecisão das medições do recetor GPS é o principal problema a afetar a qualidade imersiva. É de crer que este problema seria substancialmente atenuado no cenário de utilização de referência (*audio-guiding* em autocarros turísticos), pois este é mais previsível e envolve uma escala de distâncias bastante maior. O teste de rotas neste cenário é uma das tarefas a curto prazo. Em todo o caso, uma deteção de posição mais precisa é prioritária. Aplicar filtragem aos dados provenientes do recetor GPS não é a solução mais adequada, pois a melhoria de precisão é conseguida à custa do aumento da latência. A abordagem mais prometedora parece ser a de combinar os dados da IMU e do recetor GPS (fusão sensorial).

Podem identificar-se múltiplas pistas de melhoria (otimização, adição de novas *features*) alicerçadas no trabalho desenvolvido, do ponto de vista quer do *software* quer do *hardware*. Por exemplo, o uso da instrução de *switch* (Figura 4-5) na seleção no modo de *trigger* no bloco ‘Controlador de reprodução’ foi propositado, tendo em vista a possibilidade de adicionar outros modos de *trigger*. A interface gráfica das aplicações pode ser aperfeiçoada, recorrendo, por exemplo, à *Google Maps API*. Refira-se também a possibilidade de integrar a aplicação descrita na secção 2.2.1 (Auralização com *streaming* a partir de microfone) na aplicação de reprodução, de modo a ser seleccionável a reprodução de conteúdos pré-gravados ou de comentários feitos no momento (mantendo-se a sua associação às localizações das fontes virtuais).

Está em curso a tarefa de transpor as aplicações que implementam os vários blocos (reprodução, GPS, gestor de rota...) para *Android*, pois a estrutura do sistema será simplificada, tornando-se simultaneamente mais leve e versátil, se se concentrarem todas as funções de comunicação e processamento num *smartphone* ou *tablet* – vide Figura 1-9. Obviamente, a IMU não poderá ser incorporada, na medida em que tem que ser solidária com a cabeça do ouvinte. A integração com os auscultadores de um dispositivo de *head-tracking* miniaturizado e de baixo custo, de preferência com ligação sem fios, é uma importante frente de trabalho futuro.

Como sugere a Figura 1-9, o sistema funcionaria de forma completamente autónoma, sendo o áudio descarregado da Internet de acordo com a rota escolhida. Sob esta arquitetura, o sistema poderia adaptar-se a uma mais ampla gama de cenários, incluindo ambientes interiores. Nestes, a deteção de posição não poderia ser assegurada por GPS – seriam necessários sensores de outro tipo, por exemplo sensores de identificação por rádio-frequência (RFID), *wifi* ou faróis de ultrassons. Seria também aconselhável fazer uso das funcionalidades da biblioteca AcousticAVE para modelar a influência acústica da sala (reflexões e reverberação).

A mais-valia dos efeitos de realidade aumentada áudio aqui propostos será, sem dúvida, muito influenciada por outros fatores, como:

- Qualidade de reprodução – são necessários auscultadores de alta fidelidade e ergonómicos, bem como assentos confortáveis e com pouca vibração. O uso de sistemas de cancelamento ativo para evitar fontes de ruído externas pode revelar-se indispensável.
- Desenho de conteúdos – Explorar devidamente a dimensão 3D requer a construção de narrativas interessantes (e.g. para reconstrução histórica), baseadas em informação genuína e *design* de som profissional.
- Modelos de negócio – para afirmar este nicho de mercado, pode ser decisivo o estabelecimento de parcerias de I&D com operadores turísticos, que facilitarão, por exemplo, a criação de rotas exemplarmente desenhadas para demonstração/divulgação do conceito.



## Anexo A - Software

Foram exploradas várias opções (de ambiente de desenvolvimento de integrado) para o desenvolvimento das aplicações realizadas neste trabalho e foi escolhido o Qt Creator.

### Microsoft Visual Studio

O Microsoft Visual Studio é um IDE da Microsoft. É usado para o desenvolvimento de programas de computador, aplicações *web*, *web sites*, *web services*, e aplicações móveis. Suporta 36 linguagens de programação diferentes incluindo C/C++, C#, Java, Javascript, CSS, Node.js e Python. Permite a criação e integração de interfaces gráficas (GUI) nas aplicações.

### Eclipse

O projeto Eclipse foi em 2001 fundado pela IBM, que desenvolveu a primeira versão do produto, que depois cedeu à comunidade em geral tornando-o *free software*. Neste momento, o seu desenvolvimento é feito pela Eclipse Foundation. É um IDE vocacionado principalmente para o desenvolvimento de aplicações em linguagem Java. Contudo, através de *plugins* é possível desenvolver aplicações em outras linguagens, como C/C++, PHP, Ruby, Python, Fortran e outras. Também permite a criação e integração de GUI nas aplicações criadas por ele.

### Qt Creator

É desenvolvido pela Qt Project, fundada pela Nokia, que agora se encontra a ser liderada pela Digia's, uma subsidiária da The Qt Company. O Qt Creator é um IDE *cross-platform* (o que o torna ótimo para executar as aplicações criadas nele em outras plataformas), permite o desenvolvimento de interfaces gráficas (GUI) nas aplicações criadas muito facilmente e o desenvolvimento de aplicações em C/C++, JavaScript and QML. Também apresenta um vasto leque de bibliotecas que permitem a criação de aplicações rapidamente sem recurso a bibliotecas externas, tornando assim o desenvolvimento de aplicações bastante prático. Para ambiente Windows, este editor pode funcionar com o compilador MinGW ou com o Microsoft Visual C++ (MSVC).

Foi escolhido este IDE com o compilador MinGW para o desenvolvimento das aplicações neste trabalho. Foi usado o Qt Creator versão 5.8.0 com o MinGW 5.3.0 (Anexo A).

## MinGW

O MinGW (Minimalist GNU for Windows) é *free software*, *opensource* e trata-se de uma versão portada para Microsoft Windows de um conjunto de ferramentas GNU de ambientes Linux. O MinGW inclui adaptações para Windows do compilador GNU GCC, GNU *Binutils* (*assembler*, *linker*, e gestor de arquivos) e um conjunto importante de *headers* e bibliotecas do Windows.

Inicialmente, no desenvolvimento das aplicações foi usado o compilador MSVC; contudo este compilador conduziu a bastantes problemas na compilação das aplicações que continham a biblioteca LibAAVE (AcousticAVE). Verificou-se depois que esta biblioteca tinha sido desenvolvida em ambiente Linux e optou-se por este compilador, o MinGW, conseguindo-se assim evitar esses problemas na compilação.

## PuTTY

O PuTTY é um emulador de terminal (consola série) e uma aplicação que permite a transferência de ficheiros em redes. É *software freesource* e *opensource*. Foi desenvolvido em 1999, originalmente para ambientes Microsoft Windows sendo portado depois para vários ambientes Unix. Possibilita a comunicação com portas série e o uso de vários protocolos de rede como cliente para SCP, SSH, TCP, Telnet e rlogin.



## Anexo B - Bibliotecas e *device drivers*

### LibAAVE

A LibAAVE é uma biblioteca desenvolvida na Universidade de Aveiro em 2013 no âmbito do projeto AcousticAVE (*Auralisation Models and Applications for Virtual Reality Environments*) financiado pela Fundação para a Ciência e a Tecnologia (FCT). Trata-se de *software* de acesso aberto que serve para realizar auralização em tempo real. Foi escrita em C, sem dependência de bibliotecas externas, o que a torna bastante portátil (*cross-platform*). Na sua documentação encontra-se explicada a sua compilação para sistemas Linux. Decidiu-se incluir o código desta biblioteca nas aplicações desenvolvidas e compila-la conjuntamente com as aplicações, para facilitar eventuais alterações no contexto específico deste projeto.

### PortAudio

O PortAudio é uma biblioteca *free software, open-source* que possibilita a reprodução e gravação de conteúdos áudio. Facilita o desenvolvimento de programas áudio em C/C++ que possam ser executados em diferentes plataformas como Windows, Mac OS e Unix (*cross-platform*). Providencia uma API simples para reprodução e/ou gravação de som utilizando interfaces de bloqueio de escrita/leitura ou funções simples de *callback*. A sua documentação apresenta vários exemplos com estes métodos, o que facilita a compreensão do seu funcionamento e consequentemente a escolha do indicado para este trabalho. Para incluir esta biblioteca nas aplicações desenvolvidas é preciso compilá-la antecipadamente. O procedimento de compilação para ambiente Windows encontra-se descrito em [http://portaudio.com/docs/v19-doxydocs/compile\\_windows.html](http://portaudio.com/docs/v19-doxydocs/compile_windows.html).

### Libsndfile

A Libsndfile é uma biblioteca *free software*, criada e continuamente desenvolvida por Erik de Castro Lopo desde 1999. É uma biblioteca escrita em C que permite ler, escrever e converter facilmente uma grande variedade de formatos áudio com diversas codificações. Apresenta uma API simples que possibilita assim aos programadores abstraírem-se de detalhes. A biblioteca apresenta-se já compilada para ambientes Windows (32 e 64bits).

## PL2303 Windows driver

Este *driver* é *free software* desenvolvido pela Prolific Technology Inc. É o *software* que assegura a comunicação de baixo nível entre o recetor GPS Xucal GD75 e as aplicações desenvolvidas (o *software* de nível alto). Ele tem como função emular uma porta série (RS232) através de uma porta USB, simulando assim uma UART em RS232.

O *driver* e as suas instruções de instalação encontram-se no CD que vem com o recetor GPS mas também podem ser descarregadas da página [http://www.prolific.com.tw/US/ShowProduct.aspx?p\\_id=225&pcid=41](http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41).

## Anexo C - Conteúdo do CD

No CD que integra esta dissertação encontram-se a própria dissertação em formato *pdf*, as aplicações desenvolvidas, os projetos de Qt Creator a elas relativas, as bibliotecas e *drivers* necessários e o ficheiro de rota usado nos testes, bem como os ficheiros áudio correspondentes.

### Projetos de Qt Creator e aplicações

Na pasta “Projetos” encontram-se os projetos de Qt Creator das aplicações desenvolvidas; na pasta “Aplicações” encontram-se as aplicações desenvolvidas (já compiladas) e portanto prontas a executar (ficheiros *.exe*).

A tabela seguinte apresenta a correspondência entre projetos, aplicação e capítulo da dissertação.

Nome do projeto	Nome da aplicação	Nome (Capítulo)
<i>MicProcessingStreaming.pro</i>	<i>MicProcessingStreaming.exe</i>	Auralização e <i>streaming</i> em simultâneo com microfone (capítulo 2.2.1)
<i>FileProcessingStreaming.pro</i>	<i>FileProcessingStreaming.exe</i>	Auralização e <i>streaming</i> em simultâneo com ficheiro áudio (capítulo 2.2.2)
<i>InertialCubeTest.pro</i>	<i>InertialCubeTest.exe</i>	Aplicação para aquisição de dados do recetor GPS (capítulo 3.1.3)
<i>GPSdongleTest.pro</i>	<i>GPSdongleTest.exe</i>	Aplicação para aquisição de dados do sensor inercial (capítulo 3.2.3)
<i>RouteManager.pro</i>	<i>RouteManager.exe</i>	Gestor de rota (capítulo 4.1)
<i>ImmersiveAudioGuiding.pro</i>	<i>ImmersiveAudioGuiding.exe</i>	Sistema de reprodução (capítulo 4.2)

## **Bibliotecas e *device drivers***

Na pasta “Bibliotecas e drivers” estão as bibliotecas de vínculo dinâmico (*.dll*), do PortAudio, do Libsndfile, e o *device driver* do InertiaCube 3 necessários à execução dos projetos após compilação. Também se encontra o executável do PL2303 *Windows driver* necessário às aplicações que envolvam o uso do recetor GPS.

## **Ficheiro de rota**

Na pasta “Rota” pode-se encontrar o ficheiro de rota “*Rota-UA.txt*” e os 3 ficheiros áudio mono usado nos testes descritos na secção 5: *Acolhimento-Reitor.wav*, *Biblioteca.wav* e *Mediateca.wav*.

## Anexo D - Compilação e parâmetros configuráveis

### Configurações no Qt Creator

As aplicações desenvolvidas em Qt Creator constituem projetos (ficheiros *.pro*). Caso se pretenda editar algum dos projetos desenvolvidos (com instalação prévia do Qt Creator), deve-se abrir o ficheiro *.pro* do projeto desejado com o Qt Creator (*File*→*Open File or Project...*). Após isso, devem-se adicionar as bibliotecas no diretório de compilação, onde serão gerados vários ficheiros, incluindo o executável. Começar por compilar o projeto desejado sem as bibliotecas (*Build* → *Build Project* “Nome do projeto aberto”). Este procedimento dará um erro porém criará o diretório pretendido. O caminho desse diretório pode ser obtido no campo *Build directory* que se encontra no separador *Projects* (barra de menu do lado esquerdo na interface do Qt Creator). Esse diretório apresenta duas pastas (*Debug* e de *Release*), como os projetos estão configurados para versões *release*, é para a pasta *Release* que devem ser copiadas as bibliotecas. As tabelas seguintes descrevem os ficheiros usados nos projetos e quais devem ser usados consoante o projeto.

Ficheiro	Descrição
<i>isense.dll</i>	Biblioteca de vínculo dinâmico para o funcionamento do sensor inercial InertiaCube 3 ( <i>device driver</i> ).
<i>libsndfile-1.dll</i>	Biblioteca de vínculo dinâmico de Libsndfile.
<i>portaudio_x86.dll</i>	Biblioteca de vínculo dinâmico de PortAudio.
<i>piano.raw</i>	Ficheiro de áudio usado para auralização. (Mono, 16bits por amostra, a 44,1 KHz)

Projeto	Ficheiro
<i>MicProcessingStreaming.pro</i>	<i>portaudio_x86.dll</i>
<i>FileProcessingStreaming.pro</i>	<i>piano.raw</i> <i>portaudio_x86.dll</i>
<i>InertialCubeTest.pro</i>	<i>isense.dll</i>
<i>ImmersiveAudioGuiding.pro</i>	<i>portaudio_x86.dll</i> <i>libsndfile-1.dll</i> <i>isense.dll</i>

Depois de incluídos os respetivos ficheiro no projeto ele deverá compilar sem erros (separador ‘*Build* → *Build Project* “Nome do projeto aberto”’). Contudo para projetos que envolvam o recetor GPS deve-se instalar previamente o *PL2303 Windows driver* (vide Anexo B).

## Parâmetros configuráveis

As tabelas seguintes especificam os parâmetros que podem ser alterados nos projetos e mostram quais são utilizados em cada um (vide diretivas *#define* no início de ficheiro indicado para cada caso).

Parâmetros configuráveis	Descrição
<i>D</i>	Distância (em metros). Raio da circunferência que a fonte sonora virtual descreve em redor do ouvinte.
<i>FRAMES_PER_BLOCK</i>	Número de amostras áudio por bloco.
<i>IBUF_BLOCKS</i>	Tamanho do <i>buffer</i> de entrada (em número de blocos).
<i>OBUF_BLOCKS</i>	Tamanho do <i>buffer</i> de saída (em número de blocos).
<i>SAMPLE_RATE</i>	Frequência de amostragem (em Hertz).
<i>SAMPLE_TYPE</i>	Número de <i>bits</i> de cada amostra. 1 para <i>32bits (float)</i> , 2 para <i>16bits (short)</i> , 3 para <i>signed 8bits (char)</i> e 4 para <i>unsigned 8bits (unsigned char)</i> .
<i>GPS_READINGS</i>	Número de pares de coordenadas geográficas recebidas do recetor GPS para filtragem (tamanho da janela para <i>moving average</i> ).

Projeto	Ficheiro	Parâmetros
<i>MicProcessingStreaming.pro</i>	<i>main.cpp</i>	<i>D</i> <i>FRAMES_PER_BLOCK</i> <i>IBUF_BLOCKS</i> <i>OBUF_BLOCKS</i> <i>SAMPLE_RATE</i> <i>SAMPLE_TPE</i>
<i>FileProcessingStreaming.pro</i>	<i>main.cpp</i>	<i>D</i> <i>FRAMES_PER_BLOCK</i> <i>OBUF_BLOCKS</i> <i>SAMPLE_RATE</i> <i>SAMPLE_TYPE</i>
<i>GPSdongleTest.pro</i>	<i>mainwindow.h</i>	<i>GPS_READINGS</i>
<i>ImmersiveAudioGuiding.pro</i>	<i>audioeng.h</i>	<i>FRAMES_PER_BLOCK</i> <i>OBUF_BLOCKS</i> <i>SAMPLE_RATE</i> <i>SAMPLE_TYPE</i> <i>GPS_READINGS</i>
	<i>mainwindow.h</i>	<i>GPS_READINGS</i>
	<i>triggercontroller.h</i>	<i>GPS_READINGS</i>





## Referências

- [1] Tamo Technology HK Co. Ltd, GPS Multilingual Commentary System. Disponível em: <http://www.tamotec.com/Product//8259674739.html>. [Acedido em 15 de Abril de 2018].
- [2] toGuide, TriggerPoint Wireless. Disponível em: [http://www.toguide.pt/pt/hardware/hardware\\_show/scripts/core.htm?p=hardware&f=hardware\\_show&lang=pt&idcont=123](http://www.toguide.pt/pt/hardware/hardware_show/scripts/core.htm?p=hardware&f=hardware_show&lang=pt&idcont=123). [Acedido em 15 de Abril de 2018].
- [3] Soolai Electronics Co. Ltd, Bus Audio Guide System. Disponível em: <https://soolai.en.made-in-china.com/product/LCOndajPETVW/China-Bus-Audio-Guide-System.html>. [Acedido em 15 de Abril de 2018].
- [4] Takstar Electronic Co. Ltd, WTG-500 Tour Guide System. Disponível em: <http://www.takstar.com/en/product/detail-13-38-0-400>. [Acedido em 15 de Abril de 2018].
- [5] Mix Tech Polska, ATGS02. Disponível em: <http://www.mixtechpolska.pl/en/tour-guide-system-ATGS02.htm>. [Acedido em 15 de Abril de 2018].
- [6] Mix Tech Polska, ATGS03. Disponível em: <http://www.mixtechpolska.pl/en/tour-guide-system-ATGS03.htm>. [Acedido em 15 de Abril de 2018].
- [7] Espro Acoustiguide Group, Acoustiguide Apps. Disponível em: <http://www.acoustiguide.com/smartphone-applications>. [Acedido em 15 de Abril de 2018].
- [8] FRAMED immersive projects GmbH & Co. KG, Usomo. Disponível em: <http://usomo.de/en>. [Acedido em 15 de Abril de 2018].
- [9] G. M. Ballou, Handbook for Sound Engineers, Fourth edition ed., Burlington, USA and Oxford, UK, Elsevier, 2008.
- [10] M. Vorländer, Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality, First edition ed., Germany, Springer, 2008.
- [11] Wikipedia, “Sound localization,”. Disponível em: [https://en.wikipedia.org/wiki/Sound\\_localization](https://en.wikipedia.org/wiki/Sound_localization). [Acedido em 21 de Maio de 2017].

- [12] M. Vorländer, Physically Based Real-Time Auralization, Aachen, Germany, Aachen University, 2011.
- [13] AcousticAVE: Auralisation Models and Applications in Virtual Reality Environments. Disponível em: <https://code.ua.pt/projects/acousticave>. [Acedido em 20 de Abril de 2017].
- [14] G. Campos, P. Dias, J. Vieira, J. Santos, C. Mendonça, J. P. Lamas, N. Silva and S. Lopes, “AcousticAVE: Auralisation Models and Applications in Virtual Reality Environments” in *Proc. 8th Iberian Congress of Acoustics (Tecnicaustica)*, Murcia, Spain, Oct. 29-31, 2014.
- [15] B. Gardner and K. Martin, “HRTF Measurements of a KEMAR Dummy-Head Microphone”. MIT MediaLab, 2000. Disponível em: <http://sound.media.mit.edu/resources/KEMAR.html>. [Acedido em 16 de Abril de 2017].
- [16] V. Algazi, R. Duda and D. M. Thompson, “The CIPIC HRTF database,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New York, Oct. 2001, pp. 99-102
- [17] F. Brinkmann, A. Lindau, S. Weinzierl, G. Geissler, S. van de Par, L. Aspöck, R. Obdam, M. Vorländer, “The FABIAN head-related transfer function data base” in *FG Audiokommunikation Research Data*.
- [18] O. Warusfel, “LISTEN HRTF DATABASE”, Room Acoustics Team, IRCAM, Paris, France. Disponível em <http://recherche.ircam.fr/equipes/salles/listen/>. [Accessed April, 16, 2017].
- [19] Eclipse Foundation, Eclipse IDE. Disponível em: <https://www.eclipse.org/>. [Acedido em 20 de Abril de 2017].
- [20] Microsoft, Microsoft Visual Studio. Disponível em: <https://www.visualstudio.com>. [Acedido em Abril de 2017].
- [21] The Qt Company, Qt Creator. Disponível em: <https://www.qt.io/>. [Acedido em 20 de April de 2017].
- [22] R. Bencina, PortAudio: Portable Cross-Platform Audio I/O. <http://www.portaudio.com>. [Acedido em 20 de Abril de 2017].
- [23] P. Burk. and R. Bencina, “PortAudio Documentation”. Disponível em: <http://portaudio.com/docs/v19-doxydocs/>. [Acedido em 21 de Maio de 2017].
- [24] Wikipedia, “Global Positioning System,”. Disponível em: [https://en.wikipedia.org/wiki/Global\\_Positioning\\_System](https://en.wikipedia.org/wiki/Global_Positioning_System). [Acedido em 21 de Maio de 2017].
- [25] U. N. Department of Economic and Social Affairs, Handbook on geographic information systems and digital mapping, New York, USA, United Nations Publication, 2000.
- [26] Ishveena Singh, “How accurate is the altimeter in a GPS watch?” Disponível em: <http://geoawesomeness.com/accurate-altimeter-gps-watch/>. [Acedido em 2 de Maio de 2018].
- [27] National Marine Electronics Association, “NMEA 0183 Standard” Disponível em: [www.nmea.org/content/nmea\\_standards/nmea\\_0183\\_v\\_410.asp](http://www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp). [Acedido em 21 de Maio de 2017].

- [28] SiRF Technology, Inc, “NMEA Reference Manual”. Disponível em: <https://www.sparkfun.com/datasheets/GPS/NMEA%20Reference%20Manual-Rev2.1-Dec07.pdf>. [Acedido em 16 de Abril de 2017].
- [29] K. Betke, “The NMEA 0183 Protocol”. Disponível em: <http://www.tronico.fi/OH6NT/docs/NMEA0183.pdf>. [Acedido em 16 de Abril de 2017].
- [30] PuTTY. Disponível em: <https://www.putty.org/>. [Acedido em 9 de Abril de 2017].
- [31] M. Posch, “How To Really, Truly Use QThreads; The Full Explanation”. Disponível em: <https://mayaposch.wordpress.com/2011/11/01/how-to-really-truly-use-qthreads-the-full-explanation/>. [Acedido em 16 de Abril de 2017].
- [32] W. Wysota, “Keeping the GUI Responsive”. Disponível em: <https://doc.qt.io/archives/qc/qc27-responsive-guis.html>. [Acedido em 16 de Abril de 2017].
- [33] Weisstein, Eric W. "Moving Average." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/MovingAverage.html>. [Acedido em 16 de Abril de 2017].
- [34] Intersense. Disponível em: <http://www.intersense.com>. [Acedido em 5 de Abril de 2017].
- [35] Tekgear, InertiaCube3. Disponível em: <https://www.tekgear.com/inertiacube3.html>. [Acedido em 5 de Abril de 2017].
- [36] Development and Troubleshooting Tools, “InterSense\_SDK\_4.2381”. Disponível em: <http://www.intersense.com/pages/33/154/>. [Acedido em 10 de Março de 2017].
- [37] The Qt Company, “QGeoCoordinate Class”. Disponível em: <http://doc.qt.io/qt-5/qgeocoordinate.html#distanceTo>. [Acedido em 11 de Março de 2017].
- [38] Wikipedia, “Great-circle distance”. Disponível em: [https://en.wikipedia.org/wiki/Great-circle\\_distance](https://en.wikipedia.org/wiki/Great-circle_distance). [Acedido em 11 de Março de 2017].
- [39] Erik de Castro Lopo, Libsndfile. Disponível em: <http://www.mega-nerd.com/libsndfile/>. [Acedido em 5 de Março de 2017].
- [40] Sony, “Auscultadores ZX110”. Disponível em: [https://www.sony.pt/electronics/auscultadores-banda-cabeca/mdr-zx110-zx110a-zx110ap#product\\_details\\_default](https://www.sony.pt/electronics/auscultadores-banda-cabeca/mdr-zx110-zx110a-zx110ap#product_details_default). [Acedido em 29 de Março de 2017].